



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ**

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

# **REKONSTRUKCE DIGITÁLNÍCH FOTOGRAFIÍ S VYU- ŽITÍM HLOUBKOVÉ MAPY**

IMAGE COMPLETION USING DEPTH MAPS

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**ONDŘEJ VALEŠ**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Doc. Ing. MARTIN ČADÍK, Ph.D.**

**BRNO 2017**

**Vysoké učení technické v Brně - Fakulta informačních technologií**

Ústav počítačové grafiky a multimédií

Akademický rok 2016/2017

**Zadání bakalářské práce**

Řešitel: **Valeš Ondřej**

Obor: Informační technologie

Téma: **Rekonstrukce digitálních fotografií s využitím hloubkové mapy  
Image Completion Using Depth Maps**

Kategorie: Web

**Pokyny:**

1. Proveďte rešerši existujících metod pro rekonstrukci digitálních fotografií.
2. Navrhněte a implementujte systém pro rekonstrukci fotografií s využitím hloubkových map.
3. Demonstrujte vlastnosti systému na vhodně zvolených příkladech a testovacích sadách, získané výsledky porovnejte s existujícími metodami.
4. Dosažené výsledky prezentujte formou videa, plakátu, článku, apod.

**Literatura:**

- <http://cadik.posvete.cz/locate/>

Pro udělení zápočtu za první semestr je požadováno:

1. Body 1 a 2 zadání.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Čadík Martin, doc. Ing., Ph.D., UPGM FIT VUT**

Datum zadání: 1. listopadu 2016

Datum odevzdání: 17. května 2017

**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
Fakulta informačních technologií  
Ústav počítačové grafiky a multimédií  
602 00 Brno, Božetěchova 2



doc. Dr. Ing. Jan Černocký  
vedoucí ústavu

## Abstrakt

Cílem této práce je navrhnout postup, který bude provádět rekonstrukci digitálních fotografií na základě vyhledání podobných fotografií v datasetu, jejichž obsah je poté využit při rekonstrukci, a otestovat možnosti generování fotografií přímo z hloubkové mapy pro účely rekonstrukce. K určení podobnosti je použit deskriptor fotografií GIST nově obohacený o informaci z hloubkové mapy, který umožňuje rychlé vyhledání fotografií v datasetu a nový postup vkládání dat do rekonstruované oblasti, která je na rozdíl od existujících přístupů nejprve rozdělena na podoblasti na základě analýzy hloubkové mapy. To umožňuje řešit rekonstrukci jednotlivých podoblastí zcela odděleně a přináší zlepšení výsledku rekonstrukce, protože podoblasti jsou menší a méně komplexní. Součástí práce je klasifikace fotografií v datasetu pomocí rozšířeného deskriptoru GIST a implementace systému umožňujícího vyhledávání v datasetu na základě podobnosti a za pomoci hloubkové mapy.

## Abstract

The aim of this thesis is to design and implement application for dataset driven scene completion utilizing data from similar dataset images and test the possibility of generating data used in reconstruction directly from depth map. For scene matching in dataset novel modification of GIST descriptor including depth data is used. Furthermore, depth maps can be used to split reconstructed image into parts with similar depth, simplifying reconstruction process. Part of this thesis is also computing GIST descriptors for dataset images and implementation of tools for searching dataset for similar images using depth maps.

## Klíčová slova

digitální fotografie, rekonstrukce, hloubková mapa, deskriptor GIST, Poissonova rovnice, dataset, C++, OpenCV

## Keywords

digital image, scene completion, depth map, GIST descriptor, Poisson's equation, dataset, C++, OpenCV

## Citace

VALEŠ, Ondřej. *Rekonstrukce digitálních fotografií s využitím hloubkové mapy*. Brno, 2017. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Čadík Martin.

# Rekonstrukce digitálních fotografií s využitím hloubkové mapy

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením docenta Martina Čadíka a uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Ondřej Valeš  
4. května 2017

## Poděkování

Chtěl bych poděkovat vedoucímu mé bakalářské práce docentu Martinu Čadíkovi za jeho pomoc, kterou mi poskytl během práce na tomto tématu a při psaní technické zprávy.



# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
1.1	Analýza rekonstrukce digitálních fotografií	4
1.2	Struktura práce	5
<b>2</b>	<b>Metody rekonstrukce digitálních fotografií</b>	<b>6</b>
2.1	Metody pracující pouze s rekonstruovanou fotografií	6
2.1.1	Exemplar-Based Image Inpainting	7
2.1.2	Texture Aware Video and Image Error Concealment	8
2.1.3	PatchMatch	8
2.2	Metody s dalším zdrojem dat	9
2.2.1	Scene Completion Using Millions of Photographs	10
2.2.2	StyLit	10
2.3	Technické prostředky	11
2.3.1	GIST	11
2.3.2	Poisson blending	12
2.3.3	Alpha-expansion	13
2.3.4	Superpixels	13
<b>3</b>	<b>Využití hloubkové mapy při rekonstrukci fotografií</b>	<b>15</b>
3.1	Získání hloubkové mapy	15
3.1.1	LOCATE - Visual Localization in Natural Environments	15
3.1.2	Ostatní způsoby generování hloubkových map	16
3.2	Analýza problému v kontextu hloubkových map	17
3.3	Návrh řešení	18
3.3.1	Vyhledání pomocí deskriptoru	18
3.3.2	GIST hloubkové mapy	18
3.3.3	Mapa průměrných hloubek	19
3.3.4	Zachování absolutních hodnot i citlivosti na hrany	20
3.3.5	Určení nejlepší shody	21
3.3.6	Rozdělení fotografie podle hloubky	22
3.3.7	Proces dělení	23
3.3.8	Generování z hloubkové mapy	25
3.3.9	Implementace	26
<b>4</b>	<b>Porovnání existujících metod a metod využívajících hloubkové mapy</b>	<b>28</b>
4.1	Prostředky využívané při porovnávání výsledků	28
4.2	Výsledky testování	28
4.2.1	Vyhodnocení dotazníku	30

4.2.2	Vliv velikosti datasetu . . . . .	31
4.2.3	Sémantické chyby . . . . .	31
4.2.4	Vizuální chyby . . . . .	31
4.2.5	Porovnání metod . . . . .	32
4.3	Vliv váhy přikládané fotografii a hloubkové mapě při výběru pomocí GIST	33
<b>5</b>	<b>Závěr</b>	<b>34</b>
	<b>Literatura</b>	<b>35</b>
	<b>Přílohy</b>	<b>38</b>
<b>A</b>	<b>Testovací sada</b>	<b>39</b>
<b>B</b>	<b>Dotazník</b>	<b>41</b>
<b>C</b>	<b>Obsah DVD</b>	<b>42</b>

# Kapitola 1

## Úvod

Rekonstrukcí fotografií se rozumí doplnění části fotografie novými daty tak, aby pokud možno nebylo možné rozpoznat, že k němu došlo. Je mnoho důvodů, proč se k rekonstrukci uchýlit, můžeme chtít odstranit budovu, která narušuje hezké panorama, člověka, který se připlétl do záběru náhodou nebo jednoduše vyplnit chybějící části fotografií, u kterých došlo ke ztátě dat, ať už při přenosu nebo poškozením média, na kterém byly uloženy, příklad rekonstrukce je na obrázku 1.1.



Obrázek 1.1: Příklad rekonstrukce digitální fotografie, kdy byla ze záběru odstraněna vesnice.

Přitom je potřeba, aby byl výsledek sémanticky správný. V nejlepším případě tedy vyplnit poškozené místo tím, co na fotografii ve skutečnosti bylo, případně tím, co je budovou nebo osobou zakryto. V mnoha případech ale není přesná rekonstrukce možná jednoduše proto, že nevíme, co má na fotografii být (viz obrázek 1.2). Nakonec zůstává snaha zachovat alespoň stejnou sémantickou kategorii (doplnit panorama hor výřezem jiného záběru hor a ne částí fotografie údolí) a dodržet přibližné hranice původních objektů [20]. Dosáhnutí těchto cílů nám může umožnit právě použití hloubkové mapy, protože z ní lze čerpat důležité informace o poškozeném místě.

Dalším požadavkem je vizuální srávnost, nahrazovaná oblast musí plynule navazovat na okolí, je potřeba zachovat horizont, přechody mezi popředím a pozadím. Na hranici nesmí dojít k výrazné změně barvy nebo textury. Nedokonalosti v těchto oblastech spolehlivě prozradí, že fotografie byla upravována.

Tato práce je zaměřena na případy, kdy došlo k úplné ztrátě nebo překrytí dat (odstranění člověka nebo budovy, ztráta při přenosu, ...), nezabývá se tedy možností získání dat



Obrázek 1.2: Příklad rekonstrukce digitální fotografie, kdy byla ze záběru odstraněna horská chata. Přestože je výsledek po vizuální stránce přijatelný, není rekonstrukce ideální. Došlo totiž k sémantické chybě – chata byla nahrazena jezerem.

přímo z poškozené oblasti (fotografie v mlze, špatné rozlišení, nevhodný formát, ...). Důraz je kladen na co největší autonomnost systému, uživatel by měl do průběhu rekonstrukce zasahovat co nejméně, ideálně vůbec. Jedinou výjimku tvoří vyznačení poškozené oblasti fotografie, automatickým rozpoznáváním opravované oblasti se práce nezabývá, to by navíc v některých případech (fotografie není poškozená, pouze z ní chci odstranit např. jednu z více osob) bylo velmi komplikované.

Testování je prováděno s datasetem fotografií z horského prostředí (Alps100K [8]), protože jeho součástí je databáze syntetických hloubkových map získaná z digitálních modelů terénu. Digitální výškové modely terénu jsou k dispozici pro celou planetu, ale jejich využití v městských aglomeracích je omezené.

## 1.1 Analýza rekonstrukce digitálních fotografií

Průběh rekonstrukce můžeme rozdělit na několik etap. Nejdříve je potřeba rozpoznat, že k poškození vůbec došlo, určit jeho rozsah a označit poškozenou oblast. Nejjednodušší a nejčastěji používané řešení je přenechat tuto práci uživateli. Ten vyznačí poškozenou oblast přímo do fotografie (případně vytvoří masku definující nahrazovanou oblast). Předejde se tím zbytečnému prohledávání fotografií, které poškozené nejsou.

Navíc uživatel může chtít upravovat i fotografie, u kterých není spokojený s jejich vizuální podobou, i když k žádnému poškození ve skutečnosti nedošlo. Alternativu představuje program detekující poškození automaticky [15], to umožňuje upravovat pouze skutečně poškozené fotografie a přináší problémy s chybnou detekcí (rekonstrukce je provedena v jiné oblasti než si uživatel přeje, není kompletní, nahrazeno je zbytečně mnoho nepoškozených pixelů apod.).

Dále je potřeba prozkoumat okolí poškozené oblasti a extrahovat z něj informace potřebné pro rekonstrukci [4]. To může být provedeno převodem na jinou reprezentaci fotografie (seskupení pixelů do větších celků se kterými se potom lépe pracuje [15]), detekcí hran vstupujících do poškozené oblasti, rozpoznáním různých druhů textur [3], které se v okolí vyskytují, určením sémantického obsahu zbytku fotografie a nebo vytvořením deskriptoru [13], který zařadí fotografii do prostoru umožňujícího vyhledávání.

Tyto informace jsou potom využity v přípravě náhradních dat potřebných při rekonstrukci. Může se jednat o interpolaci hran, které rozdělí poškozenou oblast na více menších

sektorů, určení jaké textury z okolí budou použity pro vyplnění jaké části poškozené oblasti, rekonstrukce ve vnitřní reprezentaci (vyplnění oblasti na úrovni skupin pixelů) nebo prohlédání datasetu a nalezení podobných fotografií, jejichž obsah bude použit při rekonstrukci.

Vlastní rekonstrukce je provedena vložím takto připravených náhradních dat do původní fotografie nebo převodem zpět z vnitřní reprezentace. V některých případech může splývat s přípravou dat, protože poškozená oblast může být například iterativně zmenšována přímým vkádáním textur z okolí [9]. Převod z vnitřní reprezentace je proveden přiřazením barvy a textury jednotlivým skupinám pixelů. V případě použití interpolace hran jsou jednotlivé sektory vyplněny jim příslušející texturou z okolí. Data z podobných fotografií z datasetu jsou upravena [19], aby splynula s hranicí poškozené oblasti.

Nakonec, pokud to zvolený přístup umožňuje, je uživateli předloženo více možných výsledků rekonstrukce, ze kterých si může vybrat nejlepší řešení.

U různých přístupů k rekonstrukci můžeme sledovat a porovnávat jejich rychlost (může být důležité pro real-time rekonstrukční systémy), množství potřebných vstupů (stačí poškozená fotografie, je potřeba dataset, je potřeba hloubková mapa, ...) a samotnou kvalitu rekonstrukce. Ta ale může být značně subjektivní. Do kvality rekonstrukce spadá sémantická správnost a vizuální správnost.

Vizuální správnost je možné otestovat pomocí dotazníku. Pokud nebudou respondenti schopni rozlišit fotografie, které nikdy nebyly poškozeny a ty, které prošly rekonstrukcí, úroveň vizuální správnosti je dostačující. Určení míry sémantické správnosti je možné porovnáním nepoškozeného originálu a rekonstruované fotografie.

## 1.2 Struktura práce

V první části je proveden rozbor existujících přístupů k rekonstrukci digitálních fotografií a principům jejich fungování. Součástí je také kategorizace těchto přístupů a seznámení se s technickými prostředky, které jsou v průběhu řešení používány.

Další část se zabývá možnostmi využití hloubkových map při rekonstrukci digitálních fotografií. A to jak v kombinaci s již existujícími metodami, kde je snaha informace z hloubkové mapy použít pro zmírnění nedostatků těchto metod, tak jako samostatný přístup k rekonstrukci. Konkrétně se jedná o využití stávajících deskriptorů pro vyhledávání podle hloubkových map, vytvoření nového deskriptoru pro účely této práce a návrh rekonstrukce řízené hloubkovou mapou.

Poslední část je zaměřena na vyhodnocení výstupů metod z první části, určení silných a slabých stránek jednotlivých přístupů a jejich vzájemné porovnání. K tomuto účelu jsou použity (pokud jsou volně dostupné) aplikace a zdrojové kódy původních autorů nebo je provedena vlastní implementace vycházející z dokumentů popisujících jejich fungování.

## Kapitola 2

# Metody rekonstrukce digitálních fotografií

K rekonstrukci fotografií lze použít dva základní přístupy. Data v nahrazované oblasti můžeme zkusit vygenerovat pouze s použitím rekonstruované fotografie, zpravidla rozšířením textur nebo hran z blízkého okolí nahrazované oblasti [4], nebo použít další zdroj informací. Například další fotografie stejného místa nebo dataset, ve kterém jsou nalezeny vizuálně podobné scény [13]. Data z tohoto zdroje jsou potom upravena a využita při opravě původní fotografie.

- Metody pracující pouze s rekonstruovanou fotografií
  - Image inpainting
  - Syntéza textur
  - Exemplar-Based Image Inpainting
  - Texture Aware Video and Image Error Concealment
  - PatchMatch
  - Content aware fill
- Metody s dalším zdrojem dat
  - Scene Completion Using Millions of Photographs
  - StyLit: Illumination-Guided Example-Based Stylization of 3D Renderings

### 2.1 Metody pracující pouze s rekonstruovanou fotografií

Pokud pracujeme pouze s rekonstruovanou fotografií, jediná možnost jak získat informace o poškozené oblasti je z jejího okolí.

Podle způsobu výběru informací z okolí rozlišujeme dva základní přístupy k rekonstrukci a to image inpainting a syntéza textur [9].

Při využití syntézy textur je poškozená oblast vyplněna pixely vygenerovanými na základě informace o textuře z předem vybrané oblasti (při generování se kopíruje zdrojová textura s přidáním náhodné chyby). Tímto přístupem je možné rekonstruovat i velké oblasti bez ztráty kvality výsledné fotografie, jedinou podmínkou je nahrazovaná oblast tvořená uniformní texturou. Nevýhodou je potřeba předem určit zdrojovou texturu [4]. Pokud není



nahrazovaná oblast tvořena uniformní texturou, musí ji uživatel manuálně rozdělit na části a ke každé definovat zdrojovou texturu. Pokud se jedná o fotografii s více texturami, které se navzájem prolínají, proces rozdělení nemusí být triviální a v některých případech nemusí být vůbec možný.

Image inpainting využívá bezprostřední okolí. Do poškozené oblasti jsou interpolovány pixely přímo z hranice nebo z širší oblasti využívané pokročilejšími metodami [3]. Poškozená oblast je iterativně zmenšována dokud není celá vyplněna, případně dokud není dosaženo dostatečné plynulosti výsledné rekonstrukce. Při vyplňování je potřeba interpolovat hrany vcházející do oblasti, aby výsledná rekonstrukce zachovala objekty ve fotografii. Výhodou tohoto přístupu je možnost rekonstrukce oblastí s různými texturami bez nutnosti dopředu rozdělit poškozenou oblast. Image inpainting není vhodný pro velké oblasti poškození, protože kvůli snaze doplnit oblast plynule je potom výsledné doplnění rozmazané, bez detailů a textur [4].

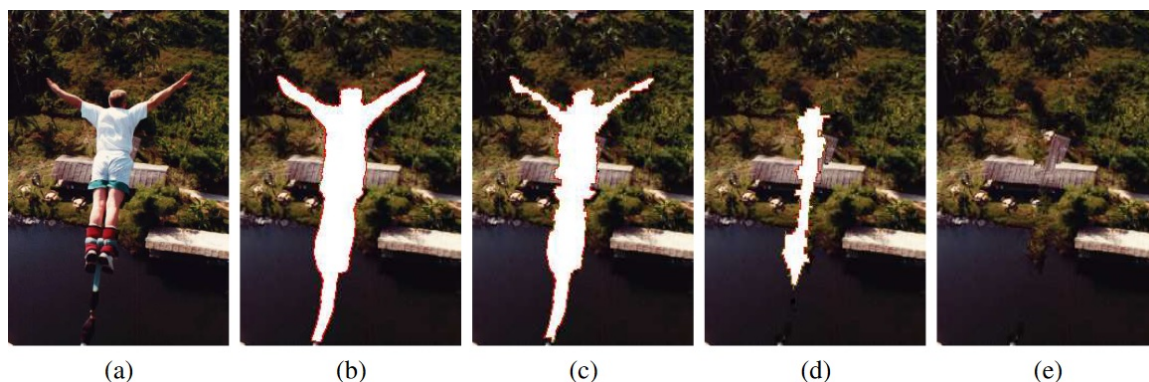
Mezi pokročilejší metody patří Texture Aware Video and Image Error Concealment [15] (popsáno v kapitole 2.1.2), která nejprve rozdělí fotografii na superpixely se stejnou texturou i obsahem a potom s nimi pracuje přístupem podobným image inpainting.

Dalším zástupcem je Exemplar-Based Image Inpainting [9] (popsáno v kapitole 2.1.1), kde se využívá image inpainting pro interpolaci hran vcházejících do poškozené oblasti. Tím je oblast rozdělena na části, které jsou potom vyplněny syntézou textur.

PatchMatch [3] (popsáno v kapitole 2.1.3) využívá rozdělení rekonstruované fotografie na políčka a jejich namapování na vizuálně nejpodobnější políčka v rámci zdrojové fotografie. Modifikaci této metody využívá také Content aware fill využívaný v aplikaci Photoshop.

### 2.1.1 Exemplar-Based Image Inpainting

Rekonstrukce kombinuje přístupy image inpainting a syntézu textur [9]. Využívá interpolaci hranic z okolí do poškozené oblasti stejně jako image inpainting. Větší oblasti zbylé po doplnění hran jsou rekonstruovány pomocí syntézy textur. Jako vzor je vybrána textura z blízkého okolí poškozené oblasti.



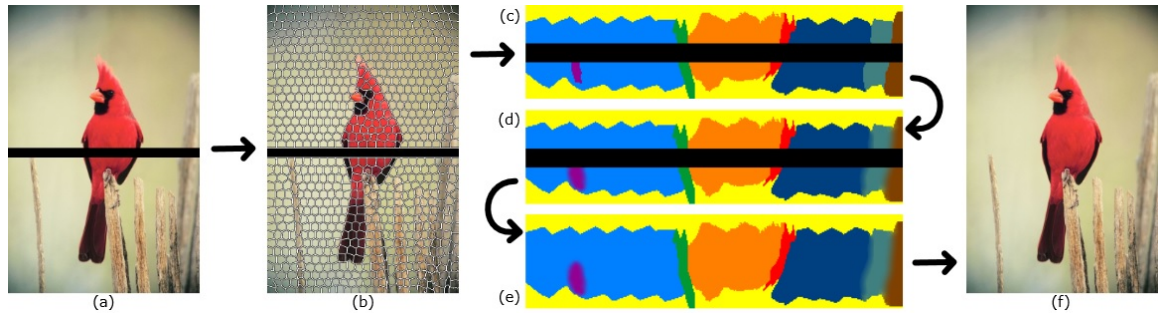
Obrázek 2.1: Průběh rekonstrukce obrázku (a) metodou Exemplar-Based Image Inpainting, postupné zmenšování oblasti (b)-(d) a výsledná opravená fotografie (e). Zdroj: Texture Aware Video and Image Error Concealment [15].

Rekostrukce postupuje iterativně (viz obrázek 2.1). V každém kroku jsou v okolí poškozené oblasti nejprve detekovány hrany a typy textur. Vyplňování poškozené oblasti je prováděno kopírováním textur z okolí. Aby byly zachovány hranice, vyplňování je vedeno ve

směru, ve kterém do poškozené oblasti vstupují hrany, které jsou doplněny nejdříve. Protože kdyby byla poškozená oblast vyplňována uniformně směrem dovnitř, z každé strany by vznikla stejně silná vrstva rozšířených textur, které by se potkaly uprostřed, a výsledek by vůbec nerespektoval rozložení objektů v původní fotografii.

### 2.1.2 Texture Aware Video and Image Error Concealment

Při rekonstrukci je využit převod do vnitřní reprezentace pomocí superpixelů (viz kapitola Superpixely 2.3.4), doplnění poškozené oblasti je provedeno v této vnitřní reprezentaci, ta je nakonec převedena do podoby výsledné fotografie (příklad na obrázku 2.2) [15].



Obrázek 2.2: Průběh rekonstrukce obrázku (a) metodou Texture Aware Video and Image Error Concealment. Rozdělením pomocí superpixelů (b), rekonstrukce ve vnitřní reprezentaci (c)-(e) a výsledná opravená fotografie (f). Zdroj: Texture Aware Video and Image Error Concealment [15].

Metodou superpixelů je okolí nahrazované oblasti rozděleno na segmenty, počet superpixelů a počet iterací je volen uživatelem. Průběh generování je upraven tak, aby superpixely neobsahovaly pixely z poškozené oblasti. Aby bylo možné popisovat i neostře hranice, pracuje tento přístup s fuzzy logikou pro určování příslušnosti pixelů k superpixelu.

Takto vzniklé segmenty jsou dále spojeny do větších celků obsahujících podobnou texturu. Tím je vytvořeno jednoduché rozdělení objektů na fotografii, využití fuzzy příslušnosti k superpixelu umožňuje vytvoření částečně se překrývajících oblastí a popis oblastí s nejasnou hranicí (pozdvolný přechod mezi dvěma barvami)

Objekty na hranici poškození mají přidělené identifikátory. Hranice objektů se stejným identifikátorem, které sousedí s poškozenou oblastí z více stran, jsou spárovány přes poškozenou oblast.

V případě objektů, které nepokračují za poškozenou oblastí, jsou jejich hranice interpolovány do poškozené oblasti tak, aby výsledná hranice měla spojitou první derivaci.

Nakonec jsou superpixelům vzniklým uvnitř poškozené oblasti přiřazeny textury z okolí příslušející objektu, do které superpixel spadá. Tak je poškozená oblast vyplněna texturou, která zachovává hranice objektů v poškozené oblasti.

### 2.1.3 PatchMatch

Metoda PatchMatch [3] rozdělí rekonstruovanou fotografii na políčka  $7px \times 7px$  a vytvoří mapování mezi nejpodobnějšími políčky.

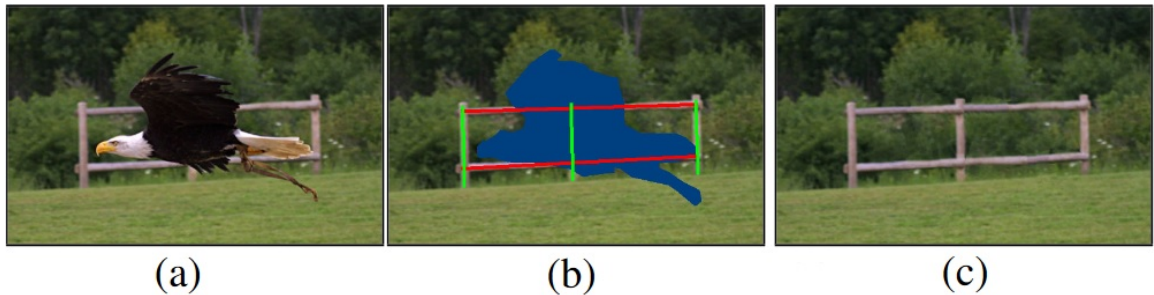
Vytváření nejlepšího mapování je výpočetně náročné, proto PatchMatch využívá lokální optimalizaci pro nalezení přibližného mapování. Počáteční mapování je vytvořeno náhodně



a vychází z předpokladu, že pro velký počet políček budou některé z těchto náhodně vytvořených mapování dostatečně kvalitní.

Potom jsou iterativně upravována mapování pro všechna políčka. Algoritmus předpokládá, že většina sousedních políček bude mít podobné mapování (políčka jsou malá, je tedy šance, že většina sousedních políček bude mít stejný nebo podobný obsah, proto i jejich nejpodobnější protějšci se budou nacházet na podobné pozici a mapování bude téměř totožné). V každé iteraci je současné mapování porovnáno s mapováním sousedních políček a několika náhodně vygenerovanými hodnotami. Jako nové mapování je vybráno to, které vede na nejpodobnější políčko. Porovnání se sousedy zaručuje propagaci dobrých hodnot z předchozí iterace a z počátečního náhodného rozdělení na sousední políčka.

Samotná rekonstrukce je potom řízena uživatelem, který ve fotografii označí poškozenou oblast a určí pomocné vodící linky. Poškozená oblast je doplňována od poškozených políček na vnějšku oblasti, ty jsou doplněny pomocí mapování sousedních nepoškozených políček. Poškozená oblast je tak vyplňována podobnými políčky, ale díky mapování se nestane, že by celá oblast byla vyplněna nakopírováním stejného políčka.



Obrázek 2.3: Příklad uživatelem řízené rekonstrukce metogou PatchMatch. Původní fotografie (a), vodící linky (b) a výsledná rekonstrukce (c). Zdroj: PatchMatch [3].

## 2.2 Metody s dalším zdrojem dat

Průběh rekonstrukce můžeme rozdělit na dvě hlavní části. První je nalezení vhodného zdroje dat (výběr z datasetu a nalezení stejné oblasti v další fotografii, vygenerování dat z hloubkové mapy), druhou úprava a vložení těchto dat do opravované fotografie.

Základním přístupem k prohledávání datasetu by byl výpočet korelačního koeficientu, ale protože pravděpodobnost nalezení dobrého zdroje dat roste s velikostí datasetu a výpočet korelačního koeficientu je časově náročný (pro velké datasety), jsou využívány deskriptory, které prohledávání značně urychlují. Pro práci s velkými datasety (milióny fotografií) je možné použít GIST deskriptor 2.3.1 a Bag of features [10]. Oba tyto přístupy zařazují položky v datasetu do  $n$ -dimenzionálního<sup>1</sup> prostoru. Podobnost se určí na základě vzdálenosti mezi deskriptorem zdroje a opravované fotografie v tomto prostoru.

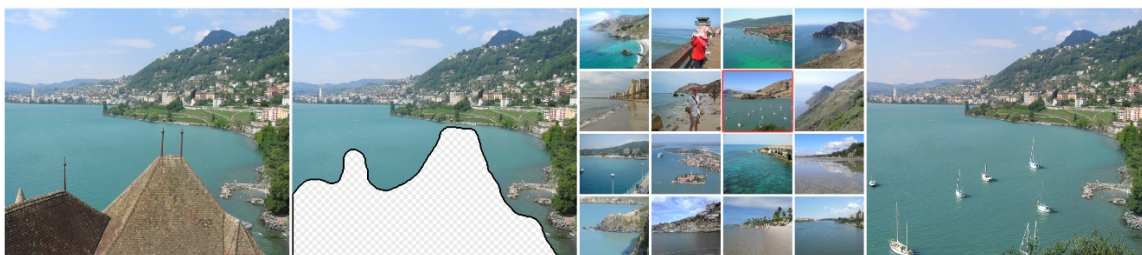
Vložení je potom možné provést přímo kopírováním, z nového zdroje použít pouze gradienty a jako integrační konstantu použít okolí z rekonstruované fotografie nebo určit divergenci, což vede k řešení Poissonovy rovnice [19] (popsáno v kapitole 2.3.2). Případně lze tyto přístupy kombinovat.

<sup>1</sup>Počet dimenzí závisí na zvolené metodě.

### 2.2.1 Scene Completion Using Millions of Photographs

První metoda využívající dataset jako další zdroj dat byla představena v práci Scene Completion Using Millions of Photographs [13]. Během rekonstrukce není vyžadován žádný zásah uživatele. Metoda produkuje více možných rekonstrukcí, ze kterých si uživatel na závěr může vybrat.

Z datasetu jsou na základě GIST deskriptoru (popsaného v kapitole GIST 2.3.1) obohaceného o barevnou informaci převzorkovanou na stejné rozlišení jako GIST vybrány kandidátní fotografie<sup>2</sup>. Přičemž váha GISTu při výběru je přibližně dvakrát větší než váha barevné informace. Tyto kandidátní fotografie by měly mít sémanticky stejný nebo podobný obsah jako rekonstruovaná fotografie.



Obrázek 2.4: Průběh rekonstrukce s použitím dalšího zdroje dat. Nejprve je označena poškozená oblast, potom jsou z datasetu vybrány podobné fotografie použité pro doplnění původní fotografie. Zdroj: Scene Completion Using Millions of Photographs [13].

V nich je potom hledána nejlepší oblast použitelná pro doplnění. Sleduje se podobnost pixelů v okolí nahrazované oblasti do vzdálenosti 80 px. Kandidátní fotografie je prohledávaná ve všech možných posuvech a ve třech měřítcích ( $1\times$ ,  $0,9\times$  a  $0,81\times$ ). V úvahu jsou brány pouze posuny, pro které se nahrazovaná oblast nedostane mimo kandidátní fotografii. Protože výběr na základě GIST by měl zaručit, že data z kandidátní fotografie jsou přibližně na správném místě, jsou velké posuny penalizovány. Využití více měřítek (zmenšení kandidátní fotografie) v podstatě ztrojnásobuje velikost zdrojového datasetu.

Pro nalezené oblasti je potom vyřešena Poissonova rovnice upraveným způsobem, který dovoluje během řešení měnit i pixely v okolí nahrazované oblasti pomocí alpha-expansion 2.3.3.

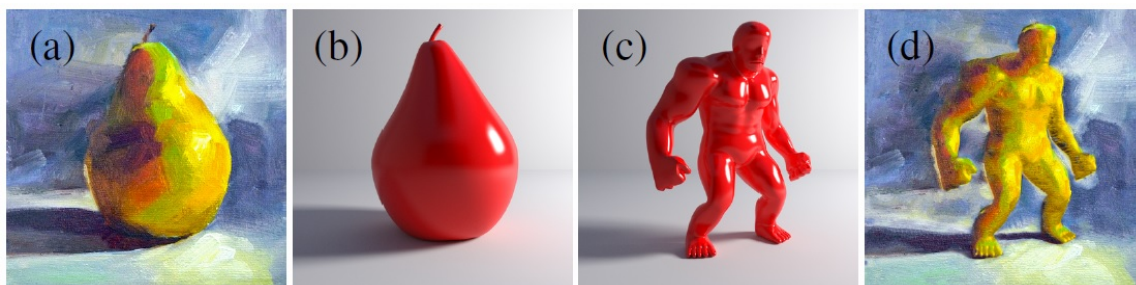
Kvalita výsledných fotografií je ohodnocena jako suma vzdálenosti v prostoru GIST a chyby vzniklé při řešení Poissonovy rovnice. Výstupem je potom 20 fotografií s nejnižším ohodnocením (viz obrázek 2.4).

### 2.2.2 StyLit

StyLit [12] není navržen pro rekonstruování digitálních fotografií, jedná se o program, který na základě modelu a barevného vzoru vytváří obarvený model. Pro účely rekonstrukce jej ale lze využít, když jako model použijeme hloubkovou mapu spolu se stínováním a k němu dodáme podobnou fotografii jako barevný vzor.

StyLit funguje na principu podobném PatchMatch. Hledá dvojice políček v modelu a zdrojové fotografii na základě podobnosti jejich stínování a hloubky. Syntetická fotografie je potom sestavena z políček zdrojové fotografie přeskládáných tak, aby co nejlépe odpovídaly modelu (viz obrázek 2.5).

<sup>2</sup>Počet kandidátních fotografií se pohybuje v rozmezí 100 až 200 snímků.



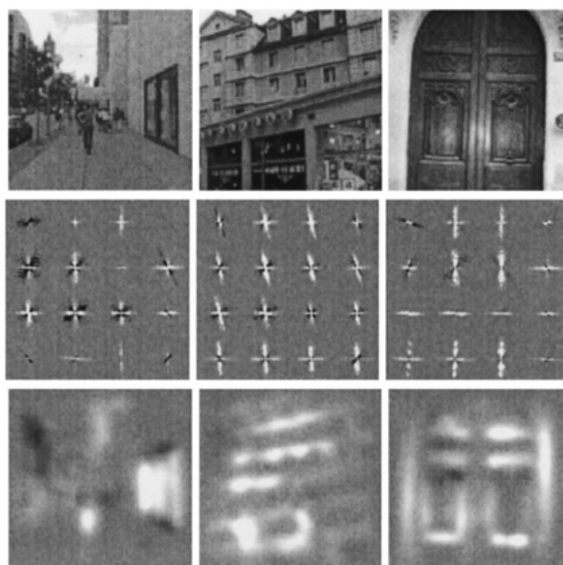
Obrázek 2.5: Příklad generování syntetických dat. Zdrojový obrázek (a) a jeho stínování (b) je využito pro obarvení modelu (c), výsledkem (d) je obarvený model. Zdroj: StyLit [13].

## 2.3 Technické prostředky

GIST je deskriptor používaný k rychlému vyhledávání podobných fotografií v rozsáhlých datsetech, který je využíván v Scene Completion Using Millions of Photographs [13] a v této práci. Vkládání dat z podobné fotografie do nahrazované oblasti vede na řešení Poissonovy rovnice. Pro nalezení nejlepší hranice, kterou obklopit nahrazovanou oblast, se používá minimalizace ceny alpha-expansion. Superpixels se používají na segmentaci fotografie na oblasti s podobnou texturou.

### 2.3.1 GIST

Jedná se o globální deskriptor, který byl původně navrhnut pro sémantickou klasifikaci fotografií [18] a úspěšně použit pro vyhledávání podobných scén za účelem rekonstrukce fotografií a odhalování plagiátů [13, 10].



Obrázek 2.6: Výpočet deskriptoru GIST fotografií (nahore), použité filtry (uprostřed) a výsledná odezva (dole). Zdroj: Evaluation of GIST Descriptors for Web-scale Image Search [18].

GIST je nízkodimenzionální reprezentace fotografie založená na určení intenzity a orientace přechodů a hran pro každý barevný kanál. Nejprve se provede konvoluce s filtrem<sup>3</sup> detekujícím hrany v různých měřících a orientacích, měřítko určuje na jak ostré nebo pozvolné přechody filtr reaguje a orientace úhel, pod kterým jsou hrany detekovány. Pro každý filtr vznikne obraz hran, ten je rozdělen mřížkou  $4 \times 4$  na 16 čtverců. Nakonec se v každém čtverci určí průměrná intenzita odezvy, která je použita jako hodnota souřadnice ve výstupním vektoru. V původní implementaci [18] byly použity šedotónové fotografie a 4 měřítka, každé po 8 orientacích, což dává výsledný vektor o délce 512 (jeden kanál, 32 kovolučních filtrů, každý obraz hran rozdělen na 16 částí). Příklad výpočtu a použitých filtrů je na obrázku 2.6.

V této práci jsou využity 4 měřítka, 8 orientací pro každou krom poslední, kde jsou pouze 4 orientace. Tato kombinace byla ukázána jako efektivní pro vyhledávání v datasetech barevných obrázků, kde není kladen důraz na detail, v práci Evaluation of GIST Descriptors for Web-scale Image Search [10].

### 2.3.2 Poisson blending

Aby byl splněn požadavek, že nahrazená oblast bude vizuálně správná a bude navazovat na okolí, je nutné vkládaná data upravit [19]. Při přímém vložení dat je jasně patrná hranice mezi původními a vloženými daty, další možností je použít gradient z vkládaných dat, což do výsledku vnáší viditelné artefakty. Jako nejlepší řešení se ukazuje použití divergence gradientů<sup>4</sup>, které popisují zřídlovost v jednotlivých barevných kanálech a vedou na řešení Poissonovy rovnice. Tento přístup splňuje požadavek plynulého přechodu na hranici oblasti a současně zachovává vizuální podobu, do které většinou nevnáší viditelné chyby. Řešení Poissonovy rovnice tedy produkuje relativně věrohodné výsledky, viz orázek 2.7.



Obrázek 2.7: Výsledek vložení dat do poškozené oblasti přímo, s použitím gradientů a po vyřešení Poissonovy rovnice.

Zřídlovost  $F_{xy}$  určíme z matice  $U'_{xy}$  obsahující vkládaná data:

$$F_{xy} = \Delta U'_{xy} = \frac{\partial^2 U'_{xy}}{\partial x^2} + \frac{\partial^2 U'_{xy}}{\partial y^2}. \quad (2.1)$$

Řešením Poissonovy rovnice se rozumí hledání takové matice  $U_{xy}$ , pro kterou platí, že hraniční hodnoty této matice jsou shodné s hodnotami na hranici nahrazované oblasti a současně splňují následující rovnici:

$$\Delta U_{xy} = F_{xy}. \quad (2.2)$$

<sup>3</sup>Gabor filter.

<sup>4</sup>Laplaceův operátor.



V některých případech nemá Poissonova rovnice řešení. Potom je za řešení považována matice  $U'_{xy}$  splňující podmínku:

$$|\Delta U_{xy} - F_{xy}| = \min |\Delta U_{xy}^* - F_{xy}|. \quad (2.3)$$

Hledáme tedy matici, která má stejnou nebo co nejpodobnější zřídlovost jako  $U'$ , ale okrajovou podmínku převzatou z původní fotografie.

K dosažení věrohodnějších výsledků je lepší nebrat zadanou masku jako neměnnou. V práci Scene Completion Using Millions of Photographs [13] je využito rozšíření masky poškození do okolí pomocí alpha expansion. Pixely jsou přiděleny buď do masky nebo mimo ni. Masku je možné pouze rozšiřovat a přidávání pixelů je penalizováno, aby se rozšířená maska velikostí příliš nelišila od té původní. Cílem je minimalizovat rozdíly v diferencích na hranici rekonstruované oblasti. Je dokázáno [13], že řešení Poissonových rovnic s takto optimalizovanou hranicí vede na plynulejší navázání doplněných dat do fotografie.

### 2.3.3 Alpha-expansion

Vyřešení Poissonovy rovnice pro náhradní data nemusí být dostačující k dosažení vizuální správnosti. Proto je vhodné rozšířit rekonstruovanou oblast takovým způsobem, aby její hranice procházela místy, kde se zdrojová data a rekonstruovaná oblast co nejvíce podobají, případně aby kopírovala hrany vyskytující se v rekonstruované fotografii. Toho je možné dosáhnout minimalizací cenové funkce pomocí rozšiřování poškozené oblasti (alpha-expansion). Při rozšiřování je každý pixel přiřazen buď k poškozené oblasti (takovýto pixel je označen jako **patch**), a bude nahrazen, nebo k původní fotografii (označen jako **exists**). Cenová funkce je:

$$C(L) = \sum_{p \in P} C_l(p, L(p)) + \sum_{p, q \in P} C_i(p, q, L(p), L(q)), \quad (2.4)$$

kde  $C(L)$  cena současného přiřazení všech pixelů,  $p$  je pixel z rekonstruované fotografie  $P$ ,  $C_l(p, L(p))$  je cena označení pixelu  $p$  jako  $L(p)$  **patch** nebo **exists**. Přiřazení pixelu z oblasti uživatelem označené k nahrazení do skupiny **exists** je silně penalizováno (označená oblast není zmenšována), stejně tak přiřazení pixelu příliš vzdáleného (více jak 80 pixelů) od oblasti uživatelem označené k nahrazení do skupiny **patch** (označená oblast není neomezeně zvětšována). Všechny ostatní případy přiřazení mají konstantní cenu, která zaručuje preferenci menšího rozšíření oblasti [13].

Druhá složka celkové ceny  $C_i(p, q, L(p), L(q))$  je cena označení dvojice pixelů. Je nulová pokud pixely patří do stejné skupiny nebo pokud spolu přímo nesousedí (počítá se tedy pouze na hranici mezi oblastí **patch** a **exists**), pro ostatní dvojice se určí jako absolutní hodnota rozdílu gradientů mezi pixely  $p$  a  $q$  v původní fotografii a vkládaných datech. Tento člen zaručuje, že hranice povede, z pohledu řešení Poissonovy rovnice, vhodnými místy.

V této práci je pro provádění alpha-expansion a rozšiřování rekonstruované oblasti využívána knihovna Multi-label optimization [6, 16, 5].

### 2.3.4 Superpixels

Segmentaci fotografie je možné provést algoritmem generujícím superpixely [1]. To jsou oblasti tvořené pixely podobné barvy, které mají v rámci každé oblasti uniformní texturu. Výsledné superpixely jsou spojitě oblasti, které mají podobnou velikost a jejich hranice kopírují hranice v původní fotografii.

Pro výpočet superpixelů je použit přístup SLIC<sup>5</sup>, jedná se o upravený *K-means clustering*, který produkuje  $k$  superpixelů. V prvním kroku jsou centra superpixelů uniformě rozmístěna po fotografii s rozestupy  $S = \sqrt{\frac{x \times y}{K}}$  a posunuty do pixelu s nejmenším gradientem v okolí  $3 \times 3$ , aby nedošlo k umístění centra na pozici kudy prochází hrana.

Potom je zahájen iterativní výpočet superpixelů, v každém kroku jsou pixely přiřazeny k nejbližšímu centru na základě barvy a fyzické vzdálenosti:

$$D = \sqrt{\left(\frac{d_c}{N_c}\right)^2 + \left(\frac{d_s}{N_s}\right)^2}, \quad (2.5)$$

kde  $d_c$  je vzdálenost v prostoru barev  $L^*a^*b^*$ ,  $d_s$  je fyzická vzdálenost pixelů a  $N_c$  a  $N_s$  jsou normalizační konstanty.

Pixel může být přiřazen pouze k centru, které leží v jeho  $2S \times 2S$  okolí.

Na závěr každé iterace jsou přepočítány centra superpixelů jako průměr pixelů náležejících k superpixelu.

Výpočet je ukončen po předem stanoveném počtu iterací nebo pokud se přestanou měnit pozice center superpixelů.

---

<sup>5</sup>Simple linear iterative clustering.

## Kapitola 3

# Využití hloubkové mapy při rekonstrukci fotografií

Informace z hloubkové mapy je možné použít ve všech fázích rekonstrukce. Rozšiřování textur z okolí můžeme omezit na pixely s podobnou hloubkou, interpolace hranic objektů do poškozené oblasti může kopírovat hranice v hloubkové mapě a nízkodimenznální popis hloubky lze zkombinovat s deskriptorem při prohledávání datasetu. Případně je možné pomocí hloubkové mapy fotografii rozdělit na sektory a rekonstrukci potom provádět v každém sektoru zvlášť.

### 3.1 Získání hloubkové mapy

Nabízí se možnost generování přímo z poškozené fotografie, to však přináší problém, protože do hloubkové mapy bude přenesena chyba z původní fotografie a pokus o použití takovéto mapy bude do řešení vnášet přesně tu chybu, kterou se snažíme odstranit.

Proto je v projektu využívám dataset s předpřipravenými hloubkovými mapami, které jsou vytvořeny na základě reálného 3D modelu Alp pomocí LOCATE<sup>1</sup> [2] (viz kapitola 3.1.1). To zaručuje, že hloubková mapa nebude obsahovat relikty objektů, které se snažíme odstranit, porovnání generování hloubkové mapy přímo z fotografie a 3D je na obrázku 3.1.

Předpoklad existence přesné hloubkové mapy ale snižuje použitelnost takovéto rekonstrukce v praxi, protože 3D model nemusí být k dispozici a vygenerování hloubkové mapy z 3D modelu vyžaduje přesné určení směru a místa, kde byla fotografie pořízena.

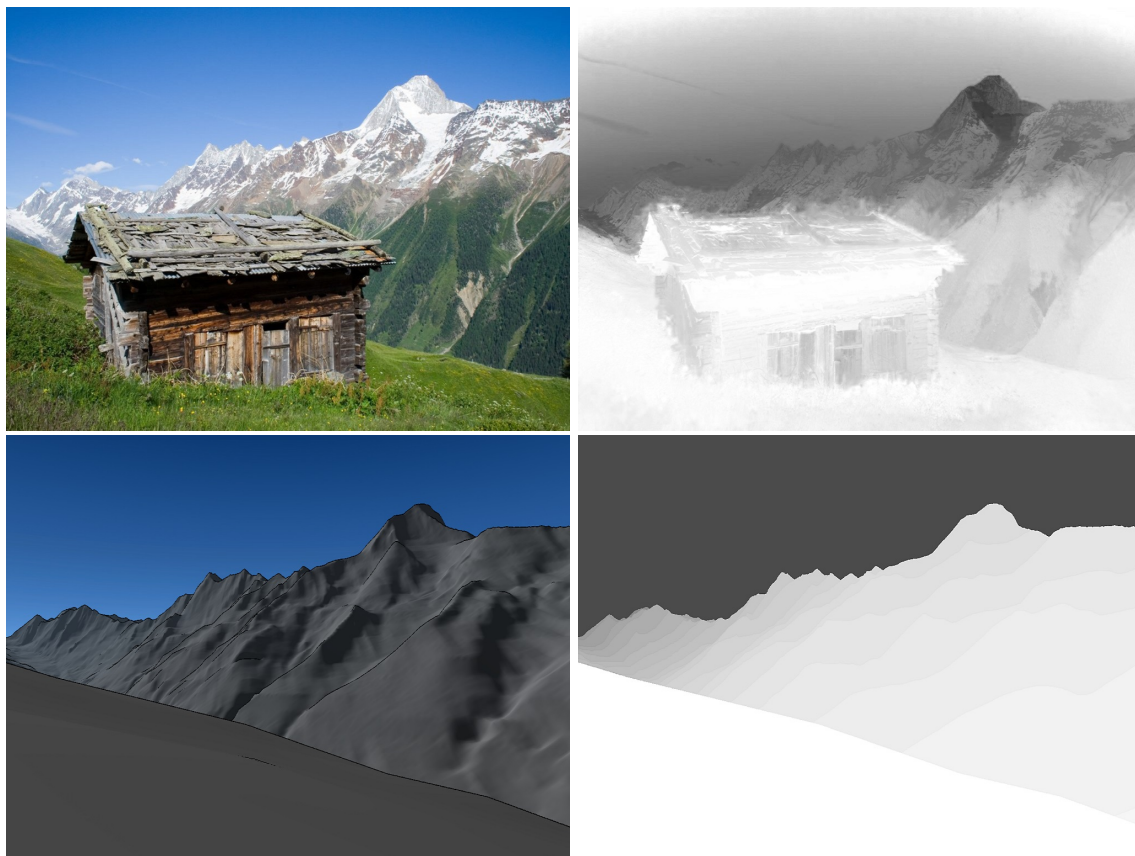
#### 3.1.1 LOCATE - Visual Localization in Natural Environments

Jednou z možností získání hloubkové mapy je zasazení fotografie do reálného modelu místa pořízení. K tomu potřebujeme přesné nalezení pozice v rámci 3D modelu a určení úhlu, pod kterým byla scéna fotografována. Potom vzdálenosti mezi místem pořízení a nejbližším objektem ve 3D modelu ve směru fotografie reprezentují hloubkovou mapu.

Projekt LOCATE [2] určuje směr pořízení na základě porovnání obsahu fotografie s referenčním modelem za předpokladu znalosti místa (GPS souřadnice zaznamenané fotoaparátem). Řešením tohoto úkolu je rotace fotografie podél všech os, jež nejvíce koreluje s modelem. Výsledkem jsou tři hodnoty úhlů, o které byla rotace provedena. Jako hodnotící kritérium se používá podobnost hran detekovaných ve vstupní fotografii a ve 3D modelu.

---

<sup>1</sup><http://cadik.posvete.cz/locate/>



Obrázek 3.1: Fotografie pro kterou se nažijme vygenerovat hloubkovou mapu (vlevo nahoře). Hloubková mapa vytvořená na základě fotografie (vpravo nahoře), na ní můžeme vidět jasné zbytky budovy, kterou se snažíme odstranit. Pohled do 3D modelu (vlevo dole). Hloubková mapa ze 3D modelu (vpravo dole).

Protože přímé nalezení výsledné rotace je výpočetně náročné, nejprve se provede omezení prohledávaného prostoru. Vstupní fotografie a množina všech možných rotací je porovnávána na souběžnost hran (oblasti, kde ve vstupní fotografii a v modelu existují rovnoběžné hrany, se na koeficientu podobnosti podílí kladně, hrany, které jsou na sebe kolmé, záporně). Výsledkem je přibližný směr porízení, správně orotovaný podél osy směru porízení. Horizont a směr “nahoru” ve fotografii odpovídá horizontu a směru “nahoru” v modelu.

V dalším kroku jsou opět porovnávány hrany ve vstupní fotografii a modelu. Tentokrát je kladen důraz na délky úseků, ve kterých jsou hrany (s jistou odchylkou) stejné. To zaručuje, že nejlepší nalezený směr nebude obsahovat velké množství křížení hran, ale co největší možnou délku vzájemně si odpovídajících hran.

### 3.1.2 Ostatní způsoby generování hloubkových map

Pokud není dostupný 3D model scény nebo je generování z modelu nepoužitelné z jiného důvodu (chybějící GPS souřadnice), můžeme zkusit hloubkovou mapu určit pouze na základě vstupní fotografie. Hloubkové mapy získané tímto způsobem ale mohou obsahovat stejnou chybu jako fotografie ze kterých byly vytvořeny.



K odhadu hloubky z jediné fotografie můžeme použít metodu Dark Channel Prior [14]. Ta vychází z předpokladu, že ve fotografii se nacházejí “tmavé” pixely s nízkou hodnotou v jednom z RGB kanálů (stíny, přirozeně tmavé objekty, objekty s barvou tvořenou jen jednou či dvěma složkami RGB, ...). Tato nejtmavší složka pixelu nesouvisí s barvou objektu, který reprezentuje, ale je vlastností celé fotografie. Jedná se o světelnost, jejíž hodnota závisí na vzálenosti pixelu od fotoaparátu a na podmínkách ve kterých byla fotografie pořízena. Čím delší trasu uzrazí světlo od foceného objektu ke kameře, tím více je pozměněno okolními vlivy. Proto i hodnoty nejtmavší složky pixelů rostou spolu s hloubkou ve fotografii. Výpočet hloubky popisuje následující rovnice:

$$t(x) = e^{-\beta d(x)} [11], \quad (3.1)$$

kde  $t(x)$  popisuje, jak se šíří světlo ve fotografii (podmínky ve kterých byla fotografi pořízena),  $d(x)$  hloubku a  $\beta$  konstanta absolutní hloubky.

Takto získaná hloubka není absolutní, pouze popisuje poměry hloubek ve fotografii (neznámá konstanta hloubky  $\beta$ ).

Další přístup řeší hledání hloubky pomocí konvoluční neuronové sítě [17]. Fotografie je nejprve rozdělena na segmety pomocí superpixelů 2.3.4, neuronová síť potom přiřazuje hloubky jednotlivým superpixelům na základě textury se zachováním vztahů mezi superpixely. Tedy sousední superpixely a superpixely s podobnou texturou mají podobnou hloubku.

## 3.2 Analýza problému v kontextu hloubkových map

Průběh rekonstrukce s využitím hloubkových map se v zásadě neliší od postupu, který se používá bez hloubkových map. Každá etapa rekonstrukce je obohacena o informace z hloubkové mapy.

Rozpoznávání poškozených oblastí můžeme obohatit porovnáním systetické a z fotografie vygenerované houbkové mapy. Oblast se zásadním rozdílem v hloubkách mezi syntetickou a generovanou mapou potom označíme za poškozenou oblast. To umožňuje (alespoň částečnou) automatizaci rozpoznávání poškození, ale detekce je omezena pouze na rozdíly mezi fotografií a 3D modelem (např. budovy a osoby v záběru) a může vést na chybnou detekci (mraky na obloze budou přeneseny do vygenerované hloubkové mapy, ale syntetické mapa je neobsahuje).

Prozkoumání okolí poškozené oblasti, sloužící k detekci hran vstupujících do poškozené oblasti a vytvoření deskriptorů popisující sémantický obsah fotografie, může být provedeno jak v rámci samotné fotografi, tak v rámci hloubkové mapy. Vyhledávání v datasetu potom používá jak deskriptor fotografie, tak hloubkové mapy.

Vložení dat do původní fotografie může být řízeno hloubkovou mapou. To zprostředkovává detekce hran v hloubkové mapě, která nahrazuje interpolaci z okolí, případně samotná hloubková mapa, kdy jsou textury kopírovány do oblastí se stejnou hloubkou.

Výjimku tvoří přístupy, které nejsou rozšířením metod pracujících bez hloubkové mapy. Do této kategorie se patří generování z hloubkové mapy 3.3.8 a doplňování po částech 3.3.6.

Při doplňování po částech nejdříve dojde k rozdělení fotografie na oblasti a pro každou oblast je potom použita libovolná metoda vyplňující jednu oblast, které jsou analyzovány v úvodu 1.1.

Generování z hloubkové mapy má průběh rekonstrukce úplně odlišný, zůstává pouze část nalezení poškozené oblasti a vložení dat. Odpadá část hledání textur a hran, která je nahrazena generátorem fotografií z hloubkové mapy.

### 3.3 Návrh řešení

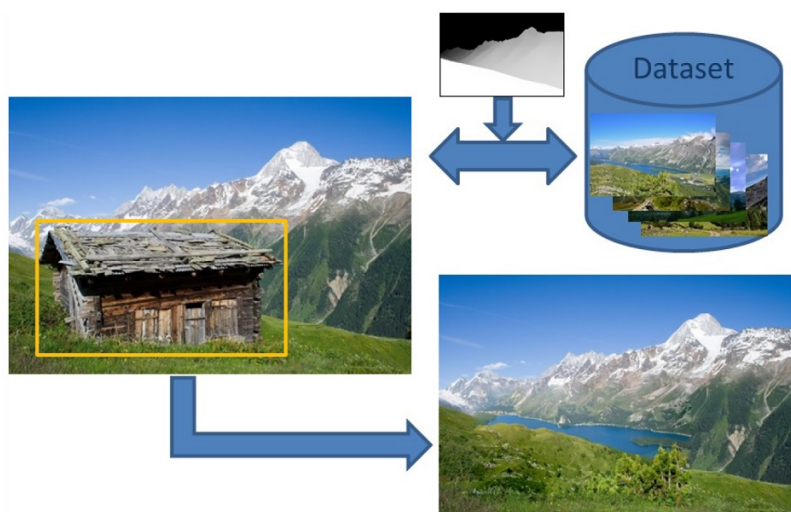
Hloubkovou mapu můžeme použít pro zlepšení výběru z datasetu 3.3.1. Dále pro určení nejlepší shody mezi rekonstruovanou fotografií a kandidáty z datasetu. Na jejím základě můžeme určit, kterou oblast z fotografie z datasetu vybrat pro nahrazení 3.3.5 a pomocí ní řídit rekonstrukci 3.3.6.

Případně lze hloubkovou mapu použít přímo k vygenerování dat, kterými bude fotografie doplněna 3.3.8.

#### 3.3.1 Vyhledání pomocí deskriptoru

První aplikace vzniklá v rámci této práce je modifikovanou verzí Scene Completion Using Millions of Photographs [13], využívá GIST hloubkové mapy 3.3.2 a mapy průměrných hloubek 3.3.3 v kombinaci s normálním vyhledáváním na základě fotografie. Uživatel připraví fotografii, kterou chce opravit, a masku definující poškozenou oblast. Dále zadá váhy, které mají být přikládány fotografii a hloubce při vyhledávání pomocí kombinovaného deskriptoru popsaného v kapitole Zachování absolutních hodnot i citlivosti na hrany 3.3.4.

V datasetu je vyhledáno 20 nejpodobnějších fotografií a zrcadlově převrácených fotografií (celkem 400 kandidátních fotografií), vyhledávání zrcadlově převrácených fotografií efektivně zdvojnásobuje velikost datasetu. Pro kandidátní fotografie je provedeno zarovnání (viz kapitola Určení nejlepší shody 3.3.5) a pro 20 fotografií s nejmenší chybou po zarovnání je provedeno rozšíření poškozené oblasti a vyřešena Poissonova rovnice (viz obrázek 3.2). Uživateli je předloženo 20 různých rekonstrukcí fotografie, kde každá varianta řešení odpovídá jedné kandidátní fotografii.



Obrázek 3.2: Ilustrace rekonstrukce pomocí kandidátní fotografie.

#### 3.3.2 GIST hloubkové mapy

Nejjednodušším způsobem využití hloubkové mapy je její převod do podoby obrázku a výpočet GIST i pro samotnou hloubkovou mapu. Protože hloubková mapa je bez textur a barev, ale reprezentuje základní rozložení scény, vyhledávání pomocí tohoto deskriptoru vrací fotografie, které jsou si podobné rozložením.

Používání pouze GISTu hloubkové mapy povede na nekvalitní výsledky, protože samotná hloubková mapa neobsahuje žádné informace o barvě. To ve výsledku způsobí sémantické chyby jako doplnění zelené louky plochou sněhu, nebo jednotnou šedou skálou a podobně.

Proto je při vyhledávání potřeba kombinovat GIST hloubkové mapy a samotné fotografie. Efektivně se tím zdvojnásobí délka deskriptoru, což vede na pomalejší vyhledávání. Výsledná podobnost dvou fotografií se potom spočítá jako:

$$D_{total} = t \times D_{photo} + (1 - t) \times D_{depth}, \quad (3.2)$$

kde  $D_{total}$  je výsledná vzdálenost,  $D_{photo}$  je vzdálenost v prostoru GIST fotografie a  $D_{depth}$  je vzdálenost v prostoru GIST. Parametr  $t$  vyjadřuje váhu kladenou barvě při určování podobnosti.

Hloubková mapa (v případě záběrů hor) bývá monotónní a oproti fotografii tvořena relativně velkými jednotnými oblastmi s pozvolně se měnícími hodnotami, má mnohem méně hran a celkově v ní jsou málo zastoupené vysoké frekvence. To vede k problémům s použitím deskriptoru GIST, protože ten je citlivý právě na hrany. Ve výsledku je vektor GIST vypočítaný z hloubkové mapy naplněn mnohem menšími hodnotami než v případě GISTu normální fotografie. Tuto odlišnost je potřeba zohlednit při volbě parametru  $t$  a na základě testování 4.3 je doporučeno volit  $t \in \langle 0, 1; 0, 4 \rangle$

Protože deskriptor GIST je sumou odezvy na filtr detekující hrany, tak nenese žádnou informaci o skutečné hloubce v jednotlivých sektorech. Přitom skutečná hloubka může být vlastnost, kterou bychom při výběru z datasetu chtěli zohlednit. Využití GISTu hloubkové mapy má v tomto směru jistou nevýhodu. Proto je v rámci této práce navrženo využívání mapy průměrných hloubek, které zachovává absolutní hodnoty hloubek.

### 3.3.3 Mapa průměrných hloubek

Mapa průměrných hloubek je pro účely této práce navrhnutý deskriptor, který je založen na absolutní hloubce fotografie. Jedná se vektor průměrných hodnot hloubky v jednotlivých oblastech fotografie.

Hloubková mapa fotografie je rozdělena na  $M \times M$  sektorů<sup>2</sup> a pro každý sektor je vypočtena průměrná hloubka. Průměrné hloubky jsou potom po řádcích umístěny do vektoru deskriptoru. Volba hodnoty  $M$  závisí na požadavcích na rychlost a kvalitu prohledání datasetu, pro velké hodnoty je výsledný deskriptor dlouhý a vyhledávání pomalé, pro malé hodnoty je zvýšena rychlost vyhledávání na úkor kvality.

Výhodou tohoto přístupu je možnost počítat průměrné hloubky jak z přeškálované šedotónové mapy, tak z originálu (obsahující floating point hodnoty) a celkově rychlejší prohledávání datasetu, protože je možné výpočty provádět ve fixed point aritmetice, neboť průměrné hloubky lze bez velké ztráty informací zaokrouhlit (což u GIST není možné).

Tento deskriptor dobře popisuje absolutní hodnoty v hloubkové mapě, stejně jako GIST hloubkové mapy však neobsahuje informace o barvě a je nutné jej kombinovat s GISTem fotografie. Protože absolutní hodnoty hloubky jsou řádově vyšší než odezva na filtr detekující hrany využitý v případě GIST, je potřeba vzdálenost v prostoru map průměrných hloubek normovat:

$$D_{total} = t \times D_{photo} + \frac{(1 - t) \times D_{avg\_map}}{N}, \quad (3.3)$$

kde  $D_{avg\_map}$  je vzdálenost v prostoru map průměrných hloubek,  $N$  je normovací konstanta a význam zbylých členů je stejný jako v případě GISTu hloubkové mapy. Parametr  $t$  je

<sup>2</sup>Pro implementaci bylo použito  $M = 32$ .

doporučeno 4.3 zvolit z intervalu  $\langle 0.4; 0.6 \rangle$ . Normalizační konstanta byla určena jako  $K = 1000000$ .

### 3.3.4 Zachování absolutních hodnot i citlivosti na hrany

Ideální způsob vyhledávání by umožňoval podle potřeby přiřazovat prioritu Jak barvě, tak hloubce a u hloubky navíc rozlišovat, jestli má být vyhledávání zaměřeno na rozložení scény (citlivost na hrany a jejich orientaci) nebo na absolutní vzdálenost částí scény. To vyžaduje zkombinování všech tří deskriptorů (GIST, GIST hloubkové mapy, mapa průměrných hloubek) do jediné výsledné vzdálenosti použité při vyhledávání:

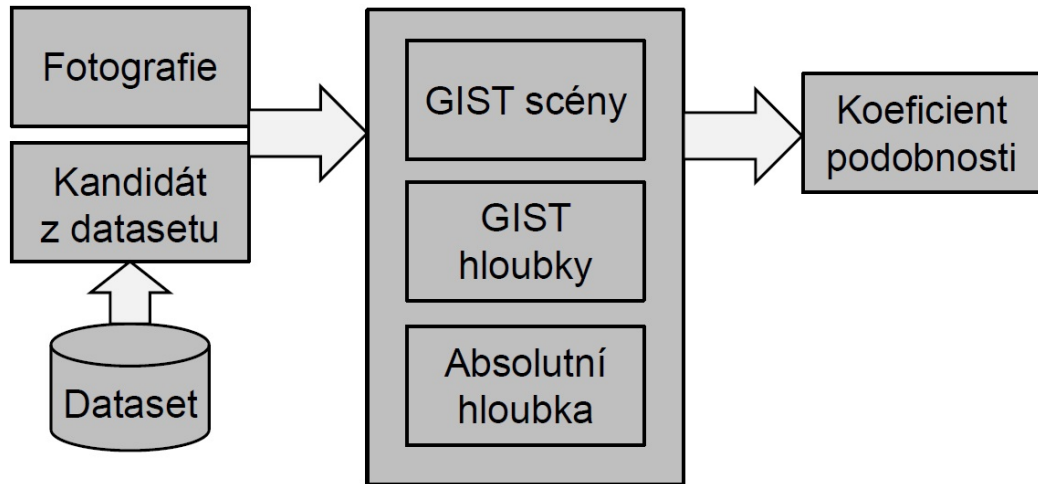
$$D_{total} = t_{color} \times D_{photo} + (1 - t_{color}) \times D_{total\_depth}, \quad (3.4)$$

kde  $t_{color}$  určuje váhu přiřazenou barvám a  $1 - t_{color}$  váhu hloubkové mapy. Člen  $D_{total\_depth}$  určující podobnost hloubkových map se vypočítá:

$$D_{total\_depth} = t_{abs} \times \frac{D_{avg\_map}}{N} + (1 - t_{abs}) \times D_{depth}, \quad (3.5)$$

$t_{abs}$  určuje váhu absolutních hodnot a  $1 - t_{abs}$  váhu rozložení scény.

Příklad nalezení podobné fotografie pomocí kombinovaného deskriptoru je na obrázku 3.3.



Obrázek 3.3: Ilustrace vyhledání podobných scén v datasetu s využitím kombinovaného deskriptoru.

Vzniká tak univerzální způsob vyhledávání kombinující jak barvu, tak hloubku, pro každé vyhledání lze navíc samostatně určit, na jaké aspekty podobnosti má být kladen důraz. Protože vyhledání kombinuje všechny tři deskriptory, je doba potřebná k vyhledání podobných fotografií delší (přibližně suma dob potřebných k vyhledání pro každý dílčí deskriptor).

Porovnání pomocí GIST je provedeno s každou fotografií v datasetu zvlášť, proto je výsledná třída časové složitosti pro jednoduchý deskriptor GIST  $O(mn)$ , kde  $m$  je délka GIST a  $n$  počet fotografií v datasetu. Kombinovaný deskriptor navržený v této práci využívá tři nezávislé deskriptory (stejně délky jako GIST), třída složitosti se proto nemění a je pouze násobena konstantou. Výsledkem je složitost  $O(3mn)$ .

Prohledání datasetu o velikosti 3000 fotografií trvá s využitím kombinovaného deskriptoru jednotky sekund, pro dataset 100 000 fotografií je to 1 až 2 minuty.

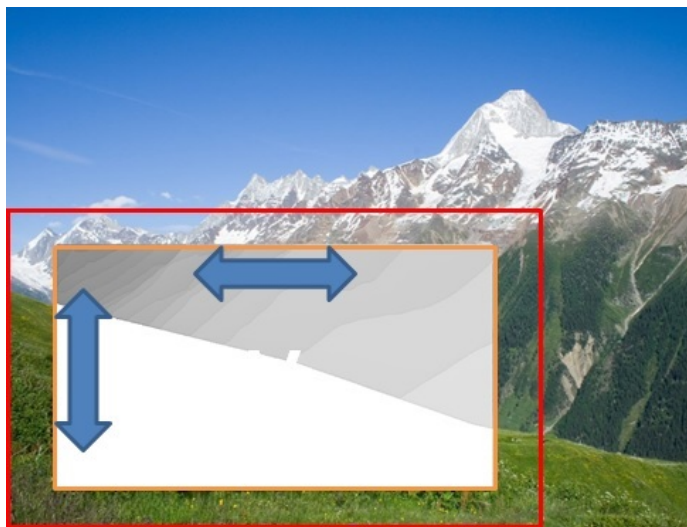
Vzniká tak univerzální způsob vyhledávání kombinující jak barvu, tak hloubku, pro každé vyhledání lze navíc samostatně určit, na jaké aspekty podobnosti má být kladen důraz. Protože vyhledání kombinuje všechny tři deskriptory, je doba potřebná k vyhledání podobných fotografií delší (přibližně suma dob potřebných k vyhledání pro každý dílčí deskriptor).

### 3.3.5 Určení nejlepší shody

Nejlepší data, která bychom použili při rekonstrukci, se v kandidátní fotografii můžou nacházet na jiné pozici než v opravované fotografii, proto je třeba každou kandidátní fotografii prohledat a nalézt nejlepší pozici shody. To se běžně dělá určením korelačního koeficientu okolí nahrazované oblasti (přitom za okolí se považují pixely do předem dané maximální vzdálenosti od hranice), který ale nevypovídá o podobnosti uvnitř nahrazované oblasti.

Ke stejnému účelu můžeme použít hloubkovou mapu, korelační koeficient se bude počítat přes celou nahrazovanou oblast v hloubkových mapách, protože hloubkovou mapu máme k dispozici i pro samu poškozenou oblast. To nám na rozdíl od výpočtu korelace okolí zaručí nejlepší přiblížení ke skutečnému obsahu poškozené oblasti. Korelace hloubkových map ale nebere v potaz rozdíl v barvách, proto je potřeba barevnou shodu zajistit jiným způsobem.

Kandidátní fotografie je prohledávána s konstantním krokem  $k = 5px$  ve směru obou os, v úvahu jsou brány pouze takové posuvy, ve kterých nahrazovaná oblast nepřesahuje hranice fotografie, okolí takto nalezené nejlepší shody je poté prohledáno s krokem  $k = 1px$ .



Obrázek 3.4: Ilustrace vyhledání nejlepší shody. V rámci oranžové oblasti (poškození) se bere v potaz rozdíl v hloubkových mapách, v rámci červené oblasti (okolí) se bere v potaz rozdíl ve fotografiích.

Možným rozšířením tohoto přístupu je prohledávání ve více měřítcích, případně přidání rotací. Tím ale roste náročnost prohledávání.

Zlepšení barevné shody lze dosáhnout zkombinováním výpočtu korelace hloubkových map v oblasti a korelace původních fotografií v okolí. Problém nastává s určením váhy, kterou přidělit oběma přístupům, protože ty budou závislé na rozsahu hodnot (fotografie



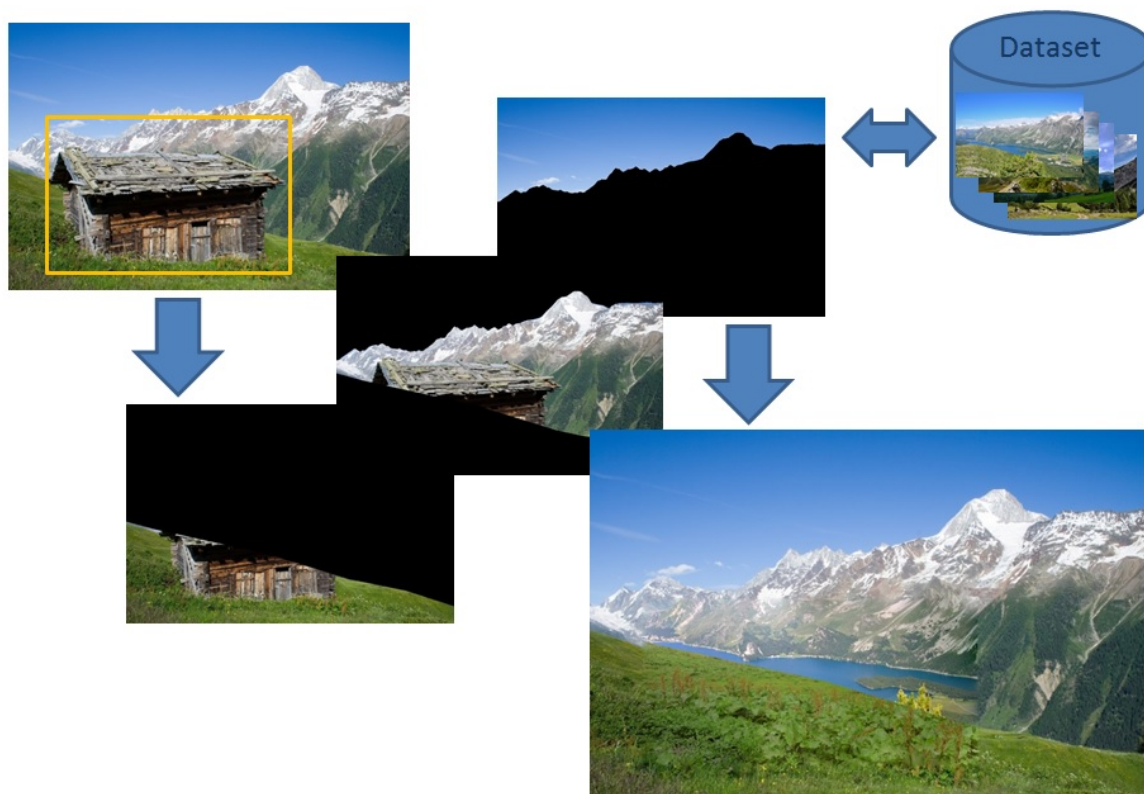
používají 8bit RGB model a hloubkové mapy 32bit floating point hodnoty), navrhovaným řešením je převést hloubkovou mapu do stejného rozsahu jako RGB kanály a přidělit jí váhu třikrát větší (má jediný barevný kanál oproti třem u RGB). Korelační koeficienty jsou potom normovány počtem pixelů, protože poměr velikosti oblasti a jejího okolí braného v potaz se může měnit (viz obrázek 3.4).

Určování nejlepší shody by mělo být provedeno až po vybrání kandidátních fotografií, protože prohledání celého datasetu tímto způsobem by bylo výpočetně náročné, a navíc pokud je použit přístup, jehož součástí je vyžití GIST deskriptoru fotografií, je tím zaručena barevná shoda.

Implementovaná varianta určování nejlepší shody kombinuje rozdíl hloubek v poškozené oblasti a koraleci okolí oblasti do vzdálenosti 80 pixelů, přiděluje jim stejnou váhu. Z důvodu náročnosti výpočtů je maximální posun omezen na 30 pixelů vertikálně i horizontálně a hodnoty obou složek nejsou normovány. Takto vzniklá hodnota je vrácena jako chyba po zarovnání.

### 3.3.6 Rozdělení fotografie podle hloubky

Protože obtížnost rekonstrukce roste se složitostí poškozené oblasti, je nutné navrhnout robustní rekonstrukční algoritmus, který se s tímto problémem vypořádá. Ostatní přístupy řeší složitost oblasti jako celek, který se snaží zjednodušit abstrakcí na superpixely, mapováním textur, případně interpolací hran. Tedy v průběhu rekonstrukce je problém postupně rozkládán na dílčí úlohy.



Obrázek 3.5: Fotografie je rozdělena na části, které jsou rekonstruovány samostatně. Výsledná složená rekonstrukce zachovává přesně původní hloubkovou mapu.

Hloubková mapa umožňuje zvolit přístup opačný. Její využití vychází z předpokladu, že související objekty se ve fotografii nacházejí přibližně ve stejné vzdálenosti. Jinak řečeno, pokud se v hloubkové mapě vyskytuje skoková změna hodnoty, pixely na jedné a druhé straně spolu nesouvisí a náleží k jinému objektu. Podél této hranice v hloubkové mapě vede ve fotografii linie přechodu mezi objekty, které na sebe navzájem vizuálně nenavazují a tato nespojitost není rušivá, protože se vyskytuje v místě, kde je očekávána. Podél těchto linií je možné celou fotografii (spolu s poškozenou oblastí) rozdělit na části, se kterými je možné pracovat odděleně. Odlišnosti vzniklé odděleným zpracováním se potom vyskytují v místě nespojitosti původní fotografie a nepůsobí rušivě (viz obrázek 3.5).

Celá rekonstrukce tak probíhá směrem od dílčích částí, které jsou rekonstruovány zvlášť (i s využitím rozdílných metod rekonstrukce), jejichž složením vznikne výsledné řešení.

### 3.3.7 Proces dělení

Od rozdělení očekáváme, že bude kopírovat hloubkovou mapu, bude generovat souvislé podoblasti (s jistou tolerancí) a počet podoblastí bude odpovídat složitosti scény. Také bychom chtěli, aby podoblasti měli definovanou minimální velikost a zabránili tak vygenerování příliš jemného dělení (aby například hory v pozadí, jež se postupně vzdalují, tvořili po rozdělení jedinou část).

K rozdělení je používána analýza histogramu hloubkové mapy. Ta nevyžaduje zadání výsledného počtu podoblastí, který tak může odrážet složitost fotografie. Umožňuje slučování sousedních intervalů, pokud podoblasti nemají minimální velikost, a na rozdíl od K-means [21] clustering nepotřebuje počáteční body v hloubkové mapě.

```

vstup : Histogram H
výstup: Vektor hraničních hodnot B
přidej dolní hranici prvního intervalu H do B;
vynuluj velikost oblasti;
foreach interval v H do
    přičti velikost intervalu k oblasti;
    if interval je lokální minimum then
        if oblast je dostatečně velká then
            přidej horní hranici intervalu do B;
            vynuluj velikost oblasti;
        end
    end
end
Vrať B;

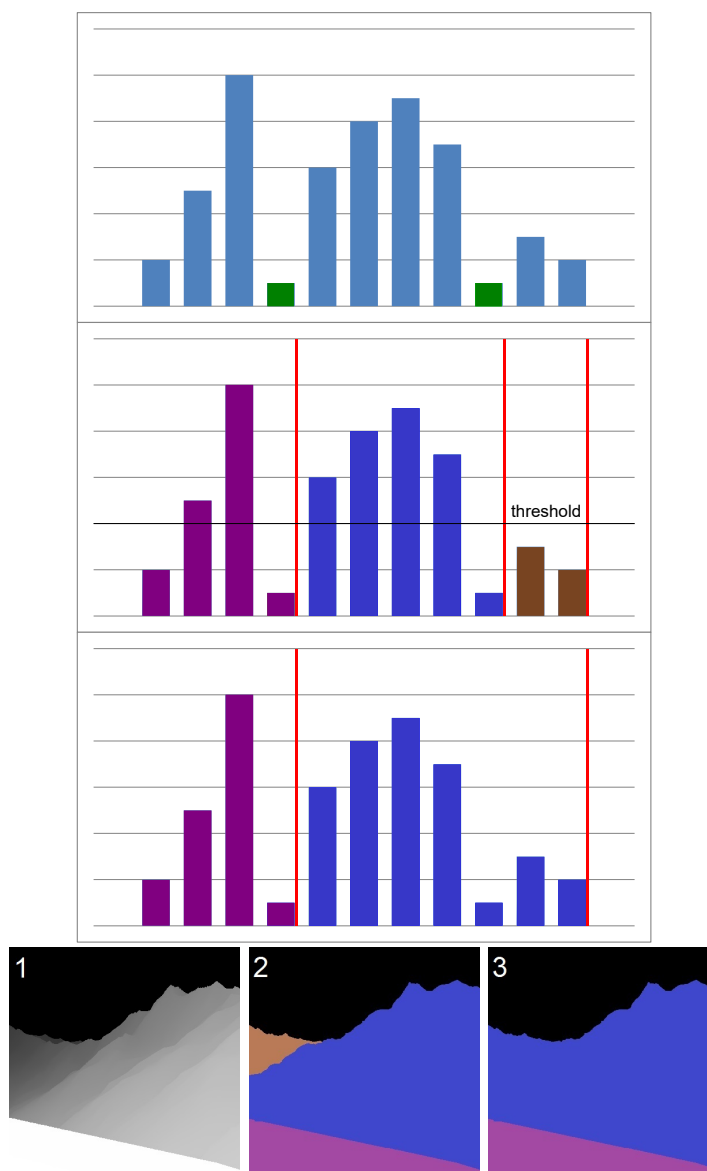
```

**Algoritmus 1:** Dělení fotografie na základě histogramu hloubkové mapy.

Hloubkové mapy mají v podstatě neomezený rozsah hodnot – od několika málo centimetrů pro objekty přímo před fotoaparátem až po nekonečno v případě oblohy. Toto je v praxi omezeno na rozsah 4 bytového floating point čísla, přesto nastává problém s volbou intervalů histogramu, který musí mít vypovídající hodnotu jak pro blízké, tak vzdálené objekty a rozumný celkový počet intervalů. Proto je využíván histogram s nesterjně velkými intervaly, kde hodnoty hranic intervalů rostou exponenciálně. To vychází z předpokladu, že s rostoucí celkovou vzdáleností nám přestává záležet na přesnosti dělení (hory na pozadí můžou být vzdálené 5000 metrů nebo 5500 metrů a pořád budou patřit ke stejnému

logickému celku, ale objekty vzdálené 10 metrů a 510 metrů už budou tvořit celky různé). Exponenciálně se zvětšující intervaly tak řeší velký rozsah hodnot a současně zachovávají potřebnou přesnost.

Pro rozsah hodnot hloubkové mapy je nejprve připraven histogram s příslušnými intervaly, který je v prvním kroku naplněn hodnotami. Poté je celý histogram rozdělen podle lokálních minim s tím, že interval, ve kterém se nachází lokální minimum, je přiřazen k sousednímu intervalu s menší hodnotou. Ve třetím kroku jsou potom nalezeny intervaly, které nesplňují podmínku minimální velikosti. Ty jsou sloučeny se sousedním intervalem stejně jako minima. Třetí krok je opakován, dokud existují intervaly s nedostatečnou velikostí. Tyto kroky je možné sloučit do jednoho algoritmu 1. Příklad rozdělení hloubkové mapy na oblasti je na obrázku 3.6.



Obrázek 3.6: Ilustrace rozdělení histogramu na části, v prvním kroku jsou nalezena minima (vyznačeno zeleně), ve druhém kroku je vytvořeno základní dělení na intervaly, ve třetím kroku je sloučen interval (označen hnědě) s nedostatečnou velikostí se svým sousedem.



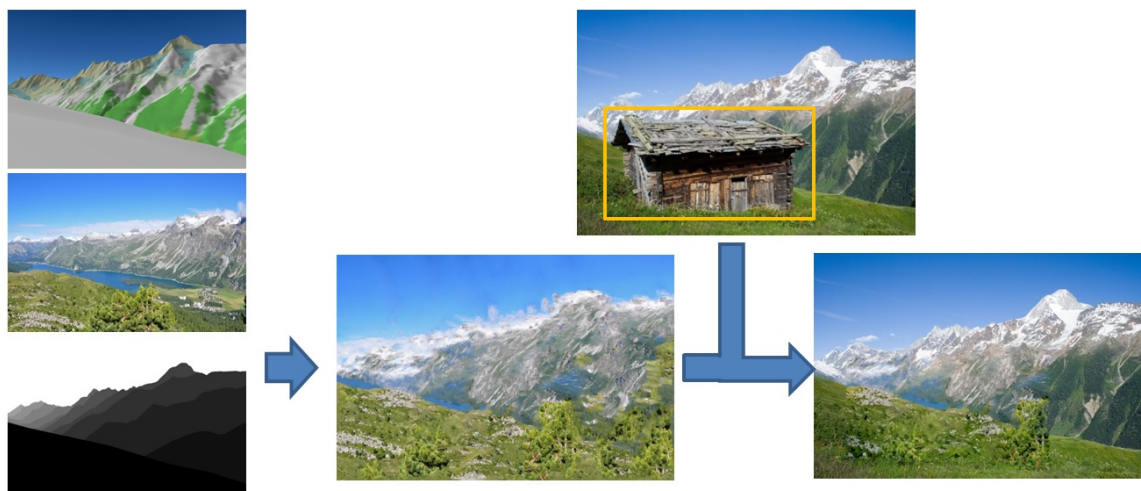
Fotografie je potom rozdělena na části podle příslušnosti hloubek do intervalů pro jednotlivé pixely.

Další možností rozdělení je K-means clustering, který iterativně seskupuje pikexy na základě jejich hodnoty, jeho nevýhodou je, že výsledný počet oblastí musí být zadán dopředu a ve výsledku tak můžeme rozdělit fotografii s relativně jednotvárnou hloubkou na zbytečně mnoho oblastí.

### 3.3.8 Generování z hloubkové mapy

Doposud jsme hloubkovou mapu využívali pro vyhledávání shody buď v rámci celého datasetu, nebo v rámci fotografie. Zcela odlišným přístupem je využít hloubkovou mapu přímo pro vygenerování dat, která potom budou vložena do poškozené oblasti.

K tomu je potřeba mít dobrou hloubkovou mapu k poškozené fotografii a jednu další fotografii, která poslouží jako zdroj textur. Fotografie sloužící jako zdroj textur může být dodána uživatelem nebo vybrána automaticky. Nabízí se použít stejný přístup jako při hledání podobných snímků v datasetu, tedy využít GIST deskriptor. Jediným rozdílem je, že fotografie z datasetu neslouží jako zdroj dat přímo, ale jsou použity jako vzor, na jehož základě jsou data vygenerována.



Obrázek 3.7: Průběh rekonstrukce s vygenerováním syntetických dat.

K vygenerování dat z hloubkové mapy je použit StyLit: Illumination-Guided Example-Based Stylization of 3D Renderings [2.2.2](#).

Před vložení vygenerovaných dat do původní fotografie může být ještě provedeno přesně zaměření a posuv, protože syntetické hloubkové mapy nemusí zcela přesně zarovnány s původní fotografií a po rekonstrukci by mohla být patrná neshoda na hranici nahrazené oblasti.

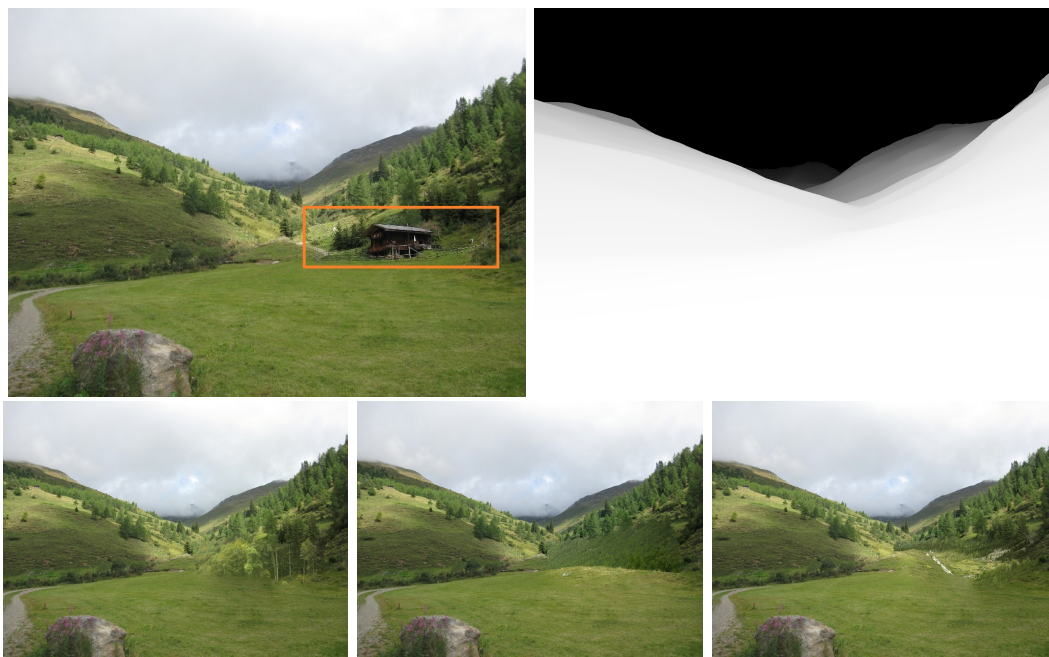
Samotná vložení dat pak probíhá stejně jako v ostatních případech, můžou být do fotografie nakopírovány přímo nebo pro ně může být vyřešena Poissonova rovnice (viz obrázek [3.7](#)).

V tomto případě je nezbytné mít opravdu kvalitní hloubkovou mapu, do které nebyla přenesena chyba způsobená poškozením fotografie. Ideální je použít syntetickou hloubkovou mapu vytvořenou na základě 3D modelu místa pořízení fotografie. Hloubková mapa vygenerovaná přímo z fotografie není pro toto využití vhodná.

### 3.3.9 Implementace

V rámci projektu byly použity datasety Alps100k [8] a jeho podmnožina GeoPose3K [7] s hloubkovými mapami. Pro účely vyhledávání byly pro celý dataset předpočítány deskriptory GIST, GIST hloubkové mapy a průměrné hloubky. Dataset není součástí řešení, jsou zahrnuty pouze skripty pro výpočty deskriptorů.

Byl implementován program **repair** pro rekonstrukci jedné fotografie s přiloženou maskou poškození a hloubkové mapy. Pro rekonstrukci je využito vyhledání podobných fotografií v datasetu, pro které uživatel volí váhy přikládané fotografii a poměrné a absolutní hloubce, jak je popsáno v kapitole Zachování absolutních hodnot i citlivosti na hrany 3.3.4. Příklad výsledků z programu **repair** je na obrázku 3.8.



Obrázek 3.8: Příklad rekonstrukce pomocí **repair**. Při prohledávání datasetu byl použit GIST hloubkové mapy a mapa průměrných hloubek. Zobrazeno několik možných výsledků rekonstrukce.

Výsledkem vyhledání je 200 nejpodobnějších fotografií, ze kterých je vybráno 10 nejlepších podle kritérií popsaných v kapitole Určení nejlepší shody 3.3.5. Z každé z nich je vytvořena rekonstruovaná fotografie vyřešením Poissonovy rovnice [19] s možnou úpravou okolních pixelů, jak je popsáno ve Scene Completion. Jako výsledek je uživateli prezentováno 10 různých verzí rekonstrukce zadané fotografie (viz obrázek 3.8).

Protože používaný dataset je relativně malý, je celá rekonstrukce zopakována s vertikálně převrácenými fotografiemi v datasetu, což efektivně zdvojnásobuje jeho velikost.

Modifikací programu **repair** je program **repair-apls**, který nemůže pracovat s hloubkovými mapami a slouží k porovnání se Scene Completion.

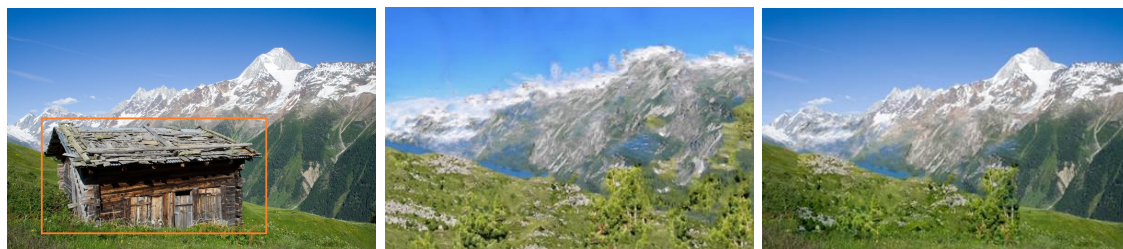
Druhý program **slice** provádí dělení podle hloubkové mapy, využívá techniku popsanou v kapitole Rozdělení fotografie podle hloubky 3.3.6. Jako základ pro exponenciální dělení intervalů používá číslo 2, které bylo vybráno na základě provedení experimentů. Po rozdělení používá pro rekonstrukci dílčích částí program **repair**, jiné metody dílčí rekonstrukce



Obrázek 3.9: Příklad rekonstrukce velké oblasti s komplexní strukturou poškozené oblasti. Fotografie byla rozdělena podle hloubkové mapy. Zobrazeny dva možné výsledky.

nebyly implementovány. Takto opravené části jsou poté bez dalších úprav složeny. Příklad výsledků z programu `slice` je na obrázku 3.9.

Poslední program `stylit` nepoužívá dataset, ale rekonstrukci provádí na základě syntetických dat vygenerovaných pomocí StyLit, jeho fungování je popsáno v kapitole Generování z hloubkové mapy 3.3.8.



Obrázek 3.10: Rekonstrukce s využitím syntetických dat (uprostřed). Výsledná rekonstrukce na fotografii vpravo.

StyLit není volně šiřitelný program, proto není využíván přímo. Pouze používá syntetické fotografie, které jsou dopředu připraveny. Ke zkvalitnění výsledků byla před generováním provedena ekvalizace histogramů stínování zdrojových dat.

Výsledná kvalita silně závisí na vygenerovaných datech, to však za běhu `stylit` nelze nijak ovlivnit. Příklad rekonstrukce pomocí `stylit` je na obrázku 3.10.

## Kapitola 4

# Porovnání existujících metod a metod využívajících hloubkové mapy

K porovnávání kvality jednotlivých metod lze přistoupit několika způsoby. Můžeme zkoumat nároky na vstupní data a techniku. Tedy výpočetní složitost použitého algoritmu, míru závislosti na vstupech uživatele (vyznačování oblastí, dodání údajů o poloze, ...) a potřeba dalších zdrojů dat (dataset, 3D model, hloubková mapa, ...).

Dále se můžeme zabývat sémantickou správností rekonstrukce, kde nás zajímá, jakými daty byla oblast doplněna, jestli není ve výsledné fotografii část oblohy vyplněná hladinou jezera, louka plochou sněhu nebo není do souvislého lesa zasazena budova.

Posledním kritériem je vizuální správnost, která je splněna při plynulém přechodu mezi originální a doplněnou částí, návazností hran a objektů. Porušení vizuální správnosti může být způsobeno i artefakty vzniklými při rekonstrukci (překročení rozsahu barev, rozmazání způsobené řešením Poissonovy rovnice, ...)

### 4.1 Prostředky využívané při porovnávání výsledků

Vnímání kvality výsledků rekonstrukce je značně subjektivní. Proto ve snaze získat alespoň částečně objektivní výsledky bude využito dotazníku.

O tom, jak moc se rekonstrukce ve fotografii projeví a jestli je poznat, že fotografie byla upravena, přináší informace hodnocení testovací skupiny. Jejím členům budou zobrazovány fotografie a jejich úkolem bude rozhodnout, jestli fotografie byla upravena nebo ne. Dotazník bude obsahovat i kontrolní neupravené fotografie.

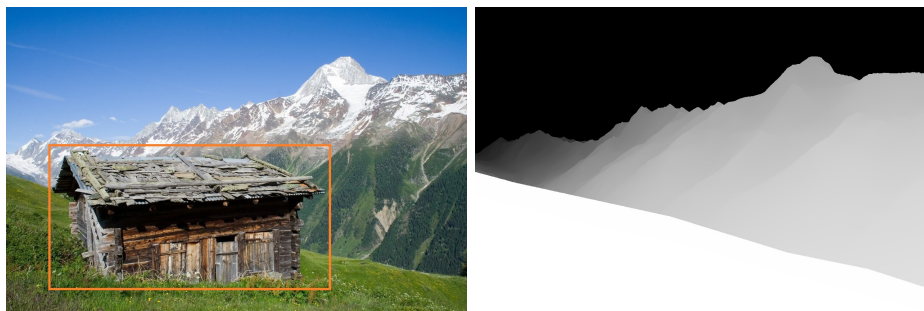
Výsledné hodnocení se bude skládat ze dvou částí: výsledků dotazníku a subjektivního hodnocení autora.

### 4.2 Výsledky testování

Testování bylo prováděno se sadou 50 fotografií (příloha A) s vyznačenými oblastmi poškození.

Při testování **repair** byly použity hodnoty  $t_{depth}$  (rovnice 3.4) 0, 3; 0, 5; a 0, 7 a  $t_{abs} = 0, 5$  (rovnice 3.5). Toto testování pracovalo s datasetem obsahujícím hloubkové mapy (GeoPose3K).



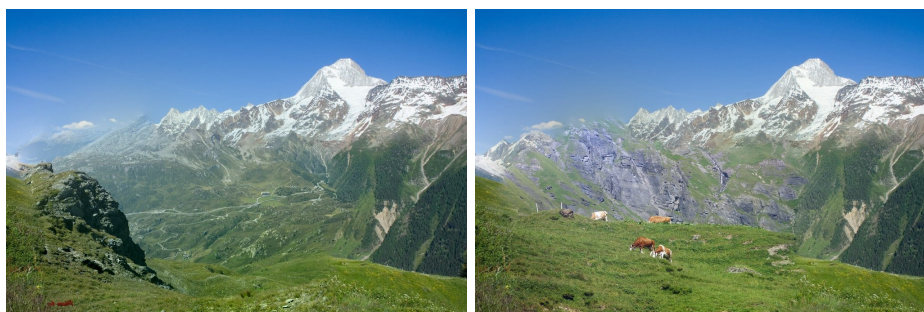


Obrázek 4.1: Rekonstruovaná fotografie a její hloubková mapa.

Pro účely porovnání byl proveden také test s  $t_{depth} = 0,0$  (bez vlivu hloubkových map) a to pro oba datasety.

Provedené testy ukazují, že vizuálně nejlepších výsledků je dosaženo s  $t_{depth} = 0,3$ . Výsledné fotografie z této sady dobře navazují na neopravované okolí a na rozdíl od rekonstrukce nevyužívající hloubkové mapy dobře zachovávají původní rozložení rekonstruované oblasti.

Příklad porovnávající výsledky je na obrázku 4.2. Fotografie na obrázku 4.2 vlevo byla rekonstruována bez využití hloubkové mapy. Po vizuální stránce je v pořádku, ale nerespektuje horizont v poškozené oblasti. Rekonstrukce fotografie na obrázku 4.2 vpravo využila hloubkovou mapu, což vedlo na lepší zachování skutečného obsahu oblasti.



Obrázek 4.2: Rekonstrukce bez hloubkové mapy (vlevo) a s hloubkovou mapou (vpravo).

V případech, kdy je na hloubkovou mapu kladen příliš velký důraz ( $t_{depth} > 0,5$ ), jsou rekonstrukce po vizuální stránce špatné, protože podobnost hloubkových map převáží podobnost samotných fotografií. To vede na výsledky, kdy je oblast doplněna daty s jinou barvou nebo texturou. Příklad rekonstrukce s příliš velkým důrazem na hloubkovou mapu je na obrázku 4.3.

Využití hloubkových map tedy přináší zlepšení výsledků ve smyslu lepšího přiblížení se původnímu obsahu. Je však závislé na barevné informaci z fotografie a pokud je informace o barvě nevyužita, rekonstrukce na základě hloubkové mapy selže.

Testování **slice** bylo provedeno se stejnou testovací sadou a datasetem GeoPose3K. Výsledky ukazují, že hloubkovou mapou řízená rekonstrukce výborně zachovává rozložení fotografie. Jediným nedostatkem je absence další úpravy dat v průběhu skládání.

Rekonstrukce pomocí slice dosahuje vysoké sémantické správnosti, která se velmi blíží skutečnému původnímu obsahu poškozeného místa. Po vizuální stránce zaostává za aplikací



Obrázek 4.3: Rekonstrukce s přílišným důrazem na hloubku.

`repair`, protože přestože se hranice oblastí nachází v očekávaných místech, stále jsou příliš výrazné. Příklad rekonstrukce pomocí `slice` je na obrázku 4.4.



Obrázek 4.4: Dva možné výsledky rekonstrukce pomocí `slice`. Oba výsledky přesně kopírují hloubkovou mapu. Na obrázku vpravo jsou jasně patrné stopy skládání dílčích oblastí.

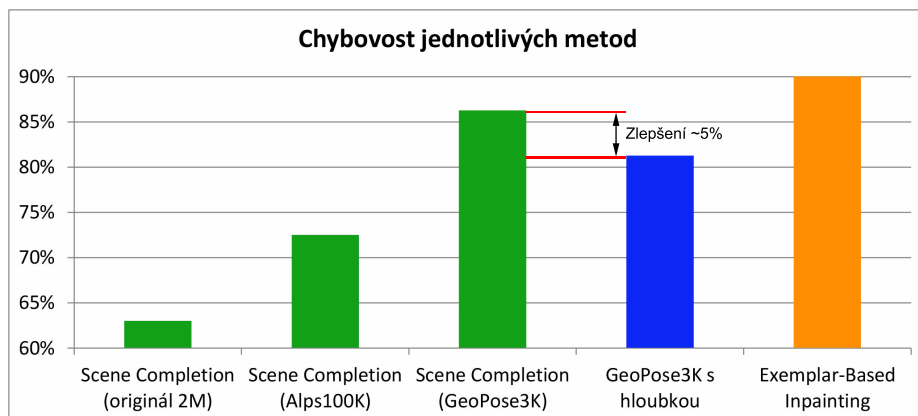
Testování `stylit` ukázalo, že v současném stavu není synteticky vygenerovaná fotografie vhodná pro účely rekonstrukce a to ani po provedení ekvalizace histogramů. Pro účely generování není v datasetu nalezena dostatečně podobná fotografie a výsledek nepůsobí přirozeně. V některých případech však lze využít i tyto data k provedení rekonstrukce s uspokojivým výsledkem. Na tuto skutečnost ale není možné spoléhat.

#### 4.2.1 Vyhodnocení dotazníku

Pro účely testování byl vytvořen dotazník obsahující 20 fotografií. U každé fotografie respondent zadával, jestli si myslí, že fotografie byla upravena nebo ne. Dotazník obsahoval originální fotografie, fotografie rekonstruované bez hloubkové mapy s využitím datasetu Alps100K, fotografie rekonstruované bez hloubkové mapy s využitím datasetu GeoPose3K [7] a fotografie rekonstruované s hloubkovou mapou s využitím datasetu GeoPose3K ( $t_{depth} = 0, 3$ ). Výsledky dotazníku jsou na obrázku 4.5.

Dotazník testoval pouze vizuální správnost, sémanticky chybné rekonstrukce (asi 10% všech výsledků) se v dotazníku nevyskytovaly. Dataset Alps100K sloužil jako kontrolní a jeho výsledky (chybovost přibližně 73%) by měly být obdobné jako u konkurenčních metod (konkrétně Scene Completion s chybovostí přibližně 63% [13]).

Tato implementace tedy nedosahuje stejné kvality jako SceneCompletion, což může být způsobeno využitím menšího datasetu a skutečností, že v této implementaci je pro řešení Poissonových rovnic a optimalizaci hranic využit software třetích stran, který není primárně určen pro tento účel.



Obrázek 4.5: Výsledky dotazníku s přidanou chybovostí Scene Completion (originál 2M) a Exemplar-Based Inpainting, převzatých z původního článku [13]. Tyto hodnoty jsou výsledkem testování s podstatně větší sadou výsledků a testovací skupinou. V grafu je vidět vzrůstající chybovost Scene Completion se zmenšujícím se datasetem (zeleně), zlepšení výsledků při použití hloubkové mapy (modře) a porovnání s metodou nevyužívající dataset (oranžově).

Zvýšení chybovosti (86%) u datasetu GeoPose3K bez hloubek je očekávané, protože tento dataset je podstatně menší. Při využití stejně velkého datasetu v kombinaci s hloubkovými mapami však chybovost klesne o přibližně 5% na 81%. Přidání informací z hloubkové mapy tedy při stejné velikosti použitého datasetu vede na zlepšení výsledků.

I s relativně malým datasetem dosahuje rekonstrukce s využitím hloubkové mapy lepších výsledků než algoritmy doplňující poškozenou oblast pouze z okolí (Exemplar-Based Inpainting s chybovostí 90%).

#### 4.2.2 Vliv velikosti datasetu

První použitý dataset Alps100k [8] obsahuje 100 tisíc fotografií, druhým datasetem je GeoPose3k obsahující 3 tisíce fotografií.

V případě většího datasetu se chybovost výsledků pohybovala okolo 73%, v případě menšího datasetu toto číslo vzrostlo o přibližně 15% (viz obrázek 4.8). Velikost datasetu má na chybovost tedy zásadní vliv, tento závěr souhlasí s testováním, které proběhlo v rámci práce Scene Completion.

#### 4.2.3 Sémantické chyby

Sémantické chyby vznikají při selhání deskriptorů při vyhledání v datasetu. Protože deskriptory používají všechny metody, četnost sémantických chyb je pro všechny varianty obdobná a pohybuje se pod 10%. Příklad sémantické chyby je na obrázku 4.6.

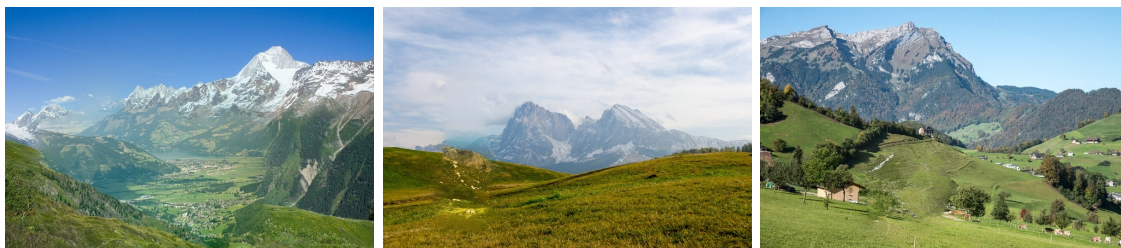
#### 4.2.4 Vizuální chyby

Při použití obou datasetů jsou velmi časté vizuální chyby, jde hlavně o nenavazující hrany mezi původní a nahrazenou částí (obrázek 4.7). Velikost datasetu nemá na četnost této chyby zásadní vliv. Tuto chybu způsobuje nalezení fotografie patřící do stejné kategorie,



Obrázek 4.6: Pokus o naharezní výseku fotografie vlevo, na obrázku vpravo výsledek rekonstrukce se sémantickou chybou. Plocha trávy byla nahrazena stromy.

ale obsahující jinou míru detailů, jinou texturu nebo jiné rozložení scény. Řešením je mnohonásobné zvětšení používaného datasetu. Autoři původní implementace [13] se se stejným problémem potýkali a zlepšení přineslo zvětšení datasetu až na velikost v řádech miliónů snímků.



Obrázek 4.7: Různé vizuální chyby.

#### 4.2.5 Porovnání metod

Na opravu úzkých poškození se nejvíce hodí jednoduché metody, protože jsou rychlé a nepotřebují dataset (Image Inpainting, Syntéza textur a Error Concealment), tyto přístupy však selhávají pro větší oblasti. Pokud je poškozená oblast větší, je potřeba využít uživatelem řízenou rekonstrukci (PatchMatch) nebo použít dataset (Scene Completion).

Aplikace **repair** je obdobou Scene Completion, pouze lépe zachovává rozložení scény, toto zlepšení je vyváženo třikrát delší dobou prohledávání datasetu a potřebou mít k dispozici hloubkové mapy. Aplikace **slice** nedosahuje po vizuální stránce tak kvalitních výsledků jako ostatní metody, její výhoda tkví ve stabilitě výsledků i pro velmi složité oblasti. Je vhodné ji použít v případě, kdy ostatní přístupy selžou. Aplikace **stylit** je vhodná pokud je k dispozici hloubková mapa pouze pro rekonstruovanou fotografii a ne pro celý dataset. Přehled metod je v tabulce 4.1.



Metoda	Výhody	Nevýhody	Vhodná pro
Error Concealment	Rychlá, autonomní	Nezachovává scénu	Úzké oblasti
PatchMatch	Rychlá, kvalitní	Řízení uživatelem	Libovolné
Scene Completion	Kvalitní, autonomní	Dataset, pomalá	Oblast s jednoduchou hranicí
repair	Zazachovává scénu	Dataset, pomalá	Oblast s jednoduchou hranicí
slice	Zazachovává scénu, kombinování přístupů	Dataset, vizuálně nekvalitní	Velmi komplexní oblasti
stylit	Nepotřebuje dataset	Potřebuje 3D model a stínování	Libovolné

Tabulka 4.1: Porovnání metod.

### 4.3 Vliv váhy přiřádané fotografii a hloubkové mapě při výběru pomocí GIST

Při výběru z datasetu je možné určovat podobnost fotografií na základě hodnoty GIST fotografie, hloubkové mapy a jejich kombinace 3.3.2.

Testování bylo provedeno rekonstrukcí testovací sady se zvětšující se hodnotou parametru  $t_{color}$  v rozsahu 0 až 1 s krokem 0,1.

Při nízkých hodnotách parametru  $t_{color}$  ( $t \leq 0,1$ ) se kandidátní fotografie shodovaly s těmi vybranými pouze na základě GISTu fotografie. Při těchto hodnotách  $t$  tedy hloubková mapa neovlivňuje výběr a tudíž se nepromítne ani do kvality výsledné rekonstrukce.

Při vysokých hodnotách  $t_{color}$  ( $t \geq 0,5$ ) se zase barvy v kandidátních fotografiích přestávají podobat rekonstruované fotografii, protože hloubková mapa žádnou informaci o barvě nenese. Vliv váhy hloubkové mapy ilustruje obrázek 4.8.



Obrázek 4.8: Rekonstrukce bez hloubkové mapy (vlevo) nerespektuje rozložení scény. Využití hloubkové mapy v malé míře (uprostřed) se lépe přibližuje očekávanému výsledku. Přílišný důraz na hloubku (vpravo) vede na vizuálně špatné doplnění.

Proto je doporučeno volit  $t_{color} \in \langle 0, 1; 0, 4 \rangle$ . To zaručuje dostačující podobnost samotných fotografií a současně zachovává informace o hloubce.

Testování prokázalo, že rozdíly mezi využíváním absolutních hloubek a GISTu hloubkové mapy jsou minimální. Proto je doporučeno volit  $t_{abs} \in \langle 0, 4; 0, 6 \rangle$  a do vyhledávání tak oba tyto aspekty promítnou s podobnou váhou.

## Kapitola 5

# Závěr

Tato práce se věnovala rekonstrukci fotografií s využitím hloubkové mapy, v jejím průběhu byly navrženy a otestovány způsoby využití hloubkové mapy při hledání v datasetu na základě absolutních hodnot hloubky, relativních poměrů v hloubkové mapě a jejich kombinace. Pro tyto účely byl využit existující deskriptor GIST a navržen nový deskriptor popisující absolutní hodnoty hloubky. Byla vytvořena aplikace implementující vyhledávání na základě hloubkových map a testování ukázalo, že při stejné velikosti datasetu vede využití hloubkových map při vyhledávání na snížení chybovosti výsledků. Lze proto předpokládat, že po zvětšení použitého datasetu dojde k překonání kvality existujícího přístupu, který hloubkové mapy nevyužívá.

Dále byla hloubková mapa použita k rozdělení komplexní rekonstruované oblasti na dílčí části, což zjednodušilo rekonstrukci rozsáhlých poškození s komplexním okolím. Hlavním přínosem je možnost využít pro rekonstrukci dílčích částí zcela libovolné metody. Výsledkem použití tohoto přístupu je přesné zachování hloubek v průběhu rekonstrukce.

Poslední zkoumaná možnost využití hloubkových map byla generování syntetických dat přímo z hloubkové mapy pomocí StyLit. Samotná rekonstrukce s využitím syntetických dat nepřináší žádné komplikace, ale kvalita rekonstrukce přímo závisí na kvalitě výsledků StyLit.

Rekonstrukce na základě hloubkové mapy tedy přináší možnost poškozené místo nahradit daty podobnými původnímu obsahu na rozdíl od konkurenčních metod, které mohou původní obsah poškozené oblasti pouze odhadovat na základě okolí. Dalším přínosem je možnost rozdělit problém rekonstrukce na podproblémy a ty řešit odděleně.

Mezi další cíle, které by měli přinést zlepšení výsledků, patří rozšíření datasetu s hloubkovými mapami, generování dostatečně kvalitních hloubkových map přímo z fotografie, přidání podpory dalších metod rekonstrukce pro dílčí oblasti po rozdělení na základě hloubkové mapy a zpracování dat v průběhu skládání dílčích částí do celku.

# Literatura

- [1] Achanta, R.; Shaji, A.; Smith, K.; aj.: SLIC Superpixels Compared to State-of-the-Art Superpixel Methods. *IEEE Trans. Pattern Anal. Mach. Intell.*, ročník 34, č. 11, Listopad 2012: s. 2274–2282, ISSN 0162-8828, doi:10.1109/TPAMI.2012.120.  
URL <http://dx.doi.org/10.1109/TPAMI.2012.120>
- [2] Baboud, L.; Čadík, M.; Eisemann, E.; aj.: Automatic Photo-to-terrain Alignment for the Annotation of Mountain Pictures. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, Washington, DC, USA: IEEE Computer Society, 2011, ISBN 978-1-4577-0394-2, s. 41–48, doi:10.1109/CVPR.2011.5995727.
- [3] Barnes, C.; Shechtman, E.; Finkelstein, A.; aj.: PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing. *ACM Trans. Graph.*, ročník 28, č. 3, Červenec 2009: s. 24:1–24:11, ISSN 0730-0301, doi:10.1145/1531326.1531330.  
URL <http://doi.acm.org/10.1145/1531326.1531330>
- [4] Bertalmio, M.; Sapiro, G.; Caselles, V.; aj.: Image Inpainting. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 2000, ISBN 1-58113-208-5, s. 417–424, doi:10.1145/344779.344972.  
URL <http://dx.doi.org/10.1145/344779.344972>
- [5] Boykov, Y.; Kolmogorov, V.: An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision. *IEEE Trans. Pattern Anal. Mach. Intell.*, ročník 26, č. 9, Září 2004: s. 1124–1137, ISSN 0162-8828, doi:10.1109/TPAMI.2004.60.  
URL <http://dx.doi.org/10.1109/TPAMI.2004.60>
- [6] Boykov, Y.; Veksler, O.; Zabih, R.: Fast Approximate Energy Minimization via Graph Cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, ročník 23, č. 11, Listopad 2001: s. 1222–1239, ISSN 0162-8828, doi:10.1109/34.969114.  
URL <http://dx.doi.org/10.1109/34.969114>
- [7] Brejcha, J.; Čadík, M.: GeoPose3K: Mountain Landscape Dataset for Camera Pose Estimation in Outdoor Environments. 2017, *Submitted to: Image Vision and Computing, Elsevier*.
- [8] Čadík, M.; Vašíček, J.; Hradiš, M.; aj.: Camera Elevation Estimation from a Single Mountain Landscape Photograph. In *British Machine Vision Conference 2015*, The

British Machine Vision Association and Society for Pattern Recognition, 2015, ISBN 1-901725-53-7, s. 1–12.

URL [http://www.fit.vutbr.cz/research/view\\_pub.php.cz?id=10930](http://www.fit.vutbr.cz/research/view_pub.php.cz?id=10930)

- [9] Criminisi, A.; Perez, P.; Toyama, K.: Region Filling and Object Removal by Exemplar-based Image Inpainting. *Trans. Img. Proc.*, ročník 13, č. 9, Září 2004: s. 1200–1212, ISSN 1057-7149, doi:10.1109/TIP.2004.833105.  
URL <http://dx.doi.org/10.1109/TIP.2004.833105>
- [10] Douze, M.; Jégou, H.; Sandhawalia, H.; aj.: Evaluation of GIST Descriptors for Web-scale Image Search. In *Proceedings of the ACM International Conference on Image and Video Retrieval*, CIVR '09, New York, NY, USA: ACM, 2009, ISBN 978-1-60558-480-5, s. 19:1–19:8, doi:10.1145/1646396.1646421.  
URL <http://doi.acm.org/10.1145/1646396.1646421>
- [11] Fattal, R.: Dehazing Using Color-Lines. *ACM Trans. Graph.*, ročník 34, č. 1, Prosinec 2014: s. 13:1–13:14, ISSN 0730-0301, doi:10.1145/2651362.  
URL <http://doi.acm.org/10.1145/2651362>
- [12] Fišer, J.; Jamriška, O.; Lukáč, M.; aj.: StyLit: Illumination-Guided Example-Based Stylization of 3D Renderings. *ACM Transactions on Graphics*, ročník 35, č. 4, 2016.
- [13] Hays, J.; Efros, A. A.: Scene Completion Using Millions of Photographs. In *ACM SIGGRAPH 2007 Papers*, SIGGRAPH '07, New York, NY, USA: ACM, 2007, doi:10.1145/1275808.1276382.  
URL <http://doi.acm.org/10.1145/1275808.1276382>
- [14] He, K.; Sun, J.; Tang, X.: Single Image Haze Removal Using Dark Channel Prior. *IEEE Trans. Pattern Anal. Mach. Intell.*, ročník 33, č. 12, Prosinec 2011: s. 2341–2353, ISSN 0162-8828, doi:10.1109/TPAMI.2010.168.  
URL <http://dx.doi.org/10.1109/TPAMI.2010.168>
- [15] Ilčíková, I.: *Texture Aware Video and Image Error Concealment*. Dizertační práce, Univerzita Komenského v Bratislave, 2016, [Online; navštíveno 29.9.2016].  
URL <http://sccg.sk/~uhlikova/files/dizertacka-final.pdf>
- [16] Kolmogorov, V.; Zabih, R.: What Energy Functions Can Be Minimized via Graph Cuts? In *Proceedings of the 7th European Conference on Computer Vision-Part III*, ECCV '02, London, UK, UK: Springer-Verlag, 2002, ISBN 3-540-43746-0, s. 65–81.  
URL <http://dl.acm.org/citation.cfm?id=645317.649315>
- [17] Liu, F.; Shen, C.; Lin, G.: Deep Convolutional Neural Fields for Depth Estimation from a Single Image. *CoRR*, ročník abs/1411.6387, 2014.  
URL <http://arxiv.org/abs/1411.6387>
- [18] Oliva, A.; Torralba, A.: Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope. *Int. J. Comput. Vision*, ročník 42, č. 3, Květen 2001: s. 145–175, ISSN 0920-5691, doi:10.1023/A:1011139631724.  
URL <http://dx.doi.org/10.1023/A:1011139631724>
- [19] Pérez, P.; Gangnet, M.; Blake, A.: Poisson Image Editing. In *ACM SIGGRAPH 2003 Papers*, SIGGRAPH '03, New York, NY, USA: ACM, 2003, ISBN 1-58113-709-5, s.

313–318, doi:10.1145/1201775.882269.

URL <http://doi.acm.org/10.1145/1201775.882269>

- [20] Pouli, T.; Reinhard, E.; Cunningham, D. W.: *Image Statistics in Visual Computing*. Natick, MA, USA: A. K. Peters, Ltd., první vydání, 2013, ISBN 978-1-56881-725-5.
- [21] Ray, S.; Turi, R. H.: Determination of number of clusters in K-means clustering and application in colour segmentation. In *The 4th International Conference on Advances in Pattern Recognition and Digital Techniques*, 1999, s. 137–143.

# Přílohy

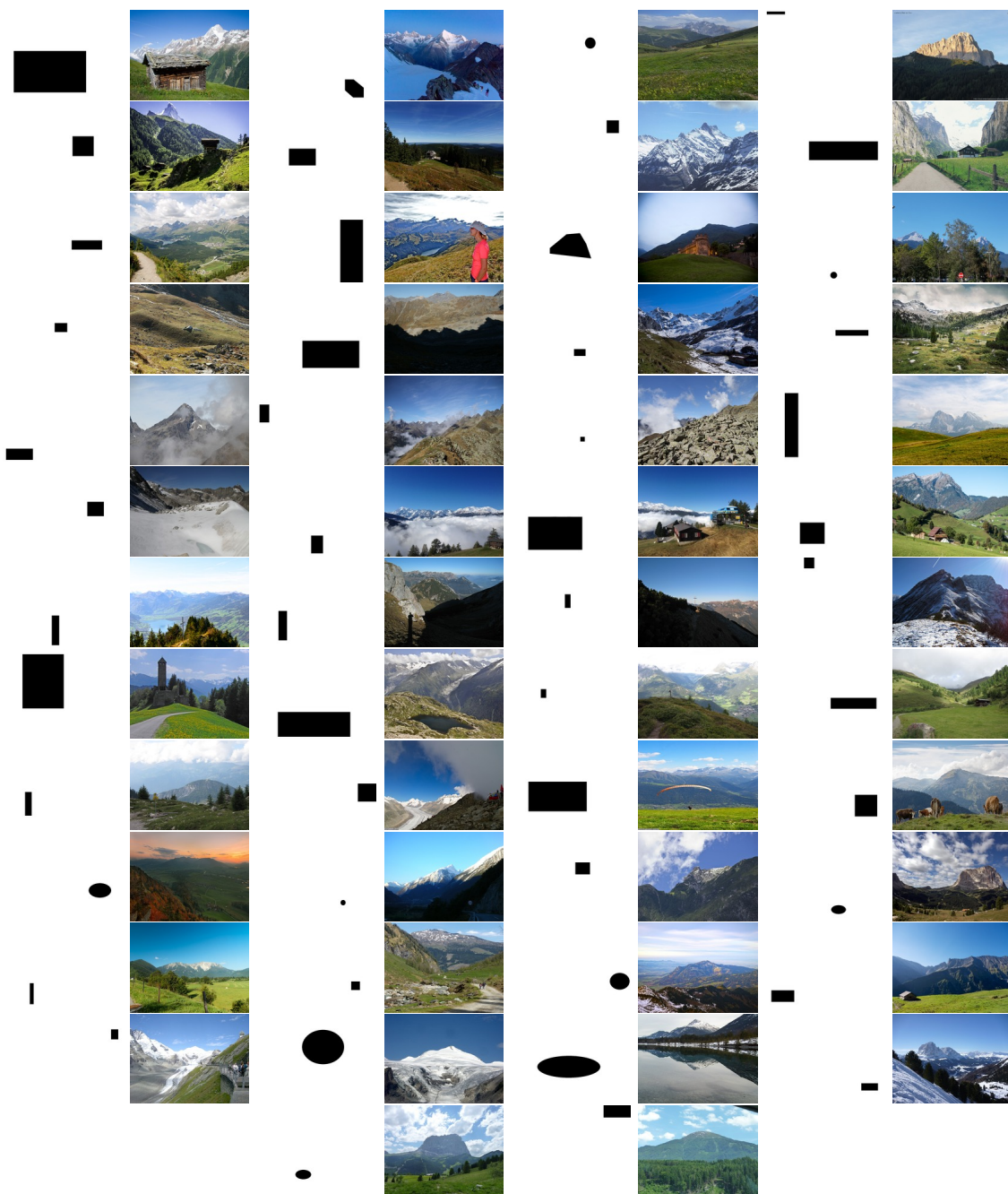


## Příloha A

### Testovací sada



Obrázek A.1: Část testovací sady ve větším rozšíření. Celá testovací sada je na obrázku [A.2](#)



Obrázek A.2: Testovací sada. Ukázka testovací sady ve větším rozlišení je na obrázku [A.1](#)

# Příloha B

## Dotazník

1/20

Obrázek 1



Je fotografie upravená? \*

☐ Určitě ano

☐ Spíše ano

☐ Spíše ne

☐ Určitě ne

Obrázek B.1: Příklad otázky v dotazníku dostupného na [https://docs.google.com/forms/d/e/1FAIpQLSfhuvbs0LVg2Co7nvYNX\\_\\_1im4Q15j8UxtnEweVph050QFKIA/viewform?usp=sf\\_link](https://docs.google.com/forms/d/e/1FAIpQLSfhuvbs0LVg2Co7nvYNX__1im4Q15j8UxtnEweVph050QFKIA/viewform?usp=sf_link)

## Příloha C

# Obsah DVD

- `app/` – adresář obsahující vytvořenou aplikaci
- `testset/` – adresář obsahující testovací sadu
- `results/` – adresář obsahující výsledky testování
- `paper/` – adresář obsahující článek, plakát a prezentaci
- `textsrc/` – zdrojové kódy k textu bakalářské práce
- `text.pdf` – text bakalářské práce